INTERNATIONAL STANDARD

ISO/IEC 29500-4

Third edition
2012-09-01

Information technology — Document description and processing languages — Office Open XML File Formats —

Part 4:
Transitional Migration Features

Technologies de l'information — Description des documents et langages de traitement — Formats de fichier "Office Open XML" —

Partie 4: Caractéristiques de migration transitoire

Reference number
ISO/IEC 29500-4:2012(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Table of Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 29500-4 was prepared by ISO/IEC JTC 1, Information technology, Subcommittee SC 34, Document description and processing languages.

This third edition cancels and replaces the second edition (ISO/IEC 29500-4:2011), which has been technically revised by incorporation of the Amendment ISO/IEC 29500-4:2011/Amd.1:2012 and the Technical Corrigendum ISO/IEC 29500-4:2011/Cor.1:2012.

ISO/IEC 29500 consists of the following parts, under the general title *Information technology — Document description and processing languages — Office Open XML File Formats*:

- *Part 1: Fundamentals and Markup Language Reference*
- *Part 2: Open Packaging Conventions*
- *Part 3: Markup Compatibility and Extensibility*
- *Part 4: Transitional Migration Features*

Annex A forms a normative part of this Part of ISO/IEC 29500. Annexes B, C, and D are for information only.

This Part of ISO/IEC 29500 includes two annexes (Annex A and Annex B) that refer to data files provided in electronic form.

The document representation formats defined by this Part are different from the formats defined in the corresponding Part of ECMA-376:2006. Some of the differences are reflected in schema changes, as shown in Annex D of this Part.

Introduction

ISO/IEC 29500 specifies a family of XML schemas, collectively called *Office Open XML*, which define the XML vocabularies for word-processing, spreadsheet, and presentation documents, as well as the packaging of documents that conform to these schemas.

The goal is to enable the implementation of the Office Open XML formats by the widest set of tools and platforms, fostering interoperability across office productivity applications and line-of-business systems, as well as to support and strengthen document archival and preservation, all in a way that is fully compatible with the existing corpus of Microsoft Office documents.

The intent of this Part of ISO/IEC 29500 is to enable a transitional period during which existing binary documents being migrated to ISO/IEC 29500 can make use of legacy features to preserve their fidelity, while noting that new documents should not use them. Part 1, §2.4, “Document Conformance”, notes that WML Strict, SML Strict and PML Strict documents do not use any of the features defined in Part 4.

This Part of ISO/IEC 29500 is normative for the current edition of ISO/IEC 29500, but is not guaranteed to be included in future revisions of that Standard. The intent is to enable the group responsible for maintenance of ISO/IEC 29500 to choose, at a later date, to remove this set of features from a revised version of that Standard.

In general, this Part of ISO/IEC 29500 augments Part 1, and inherits the provisions of that Part. Exceptions to this are indicated explicitly.

The following organizations have participated in the creation of ISO/IEC 29500 and their contributions are gratefully acknowledged:

Apple, Barclays Capital, BP, The British Library, Essilor, Intel, Microsoft, NextPage, Novell, Statoil, Toshiba, and the United States Library of Congress.

Information technology — Document description and processing languages — Office Open XML File Formats

Part 4:
Transitional Migration Features

1. Scope

ISO/IEC 29500 defines a set of XML vocabularies for representing word-processing documents, spreadsheets and presentations. On the one hand, the goal of ISO/IEC 29500 is to represent faithfully the existing corpus of word-processing documents, spreadsheets and presentations that have been produced by Microsoft Office applications (from Microsoft Office 97 to Microsoft Office 2008, inclusive). It also specifies requirements for Office Open XML consumers and producers. On the other hand, the goal is to facilitate extensibility and interoperability by enabling implementations by multiple vendors and on multiple platforms.

This Part of ISO/IEC 29500 defines features for backward-compatibility and that are useful for high-quality migration of existing binary documents to ISO/IEC 29500. These features are used only by documents of conformance class WML Transitional (§2.1), SML Transitional (§2.1), or PML Transitional (§2.1). These features are sometimes needed for high-quality migration of existing binary documents to ISO/IEC 29500.

2. Conformance

2.1 Document Conformance

A document of conformance class Office Open XML Transitional shall be a package of conformance class OPC, as specified in ISO/IEC 29500-2, for which all the following shall hold:

- The document obeys all constraints specified in this Part of ISO/IEC 29500
- The document is of category Wordprocessing, Spreadsheet, or Presentation. These categories are defined in ISO/IEC 29500-1:2011 §4
- VML Drawing Parts (§8.1) are of conformance class MCE, as specified in ISO/IEC 29500-3. Any child elements of the root element of VMLDrawing Parts are valid against the VML schema shown in A.6, "VML", after the removal of any extensions specified using the mechanisms in ISO/IEC 29500-3. VML Drawing Parts obey all constraints specified in this Part of ISO/IEC 29500
- For each OPC Part of the document of the types listed in §9.1 or ISO/IEC 29500-1:2011 §11.3, §12.3, §13.3, §14.2, and §15.2, all the following shall hold:

    i. The part is of conformance class MCE, as specified in ISO/IEC 29500-3

    ii. After the removal of any extensions using the mechanisms in ISO/IEC 29500-3, the part is valid against the Transitional W3C XML Schema (Annex A)

This Part of ISO/IEC 29500 uses the following further terms to refer to documents of conformance class Office Open XML Transitional:

- *WML Transitional*, if the document is of category Wordprocessing
- *SML Transitional*, if the document is of category Spreadsheet
- *PML Transitional*, if the document is of category Presentation

2.2 Application Conformance

Application conformance incorporates both syntax and semantics.

- A conforming consumer shall not reject any conforming documents of at least one document conformance class.
- A conforming producer shall be able to produce conforming documents of at least one document conformance class.
- A conforming application shall treat the information in Office Open XML documents in a manner consistent with the semantic definitions given in ISO/IEC 29500. An application's intended behavior need not require that application to process all of the information in an Office Open XML document. However, the information that it does process shall be processed in a manner that is consistent with the semantic definitions given in ISO/IEC 29500.

[*Note*: This note illustrates the third bullet above. Conforming applications might serve various functions. Examples include a viewer, an editor, and a back-end processor. Here is an illustration of how the third bullet applies to each of those examples:

- If a conforming viewer supports a given feature, then when it displays information using that feature, it respects the semantics of that feature as described in the Standard.
- If a conforming editor supports a given feature, then when it provides its user with an interface for manipulating information using that feature, it respects the semantics of that feature as described in the Standard.
- If a conforming back-end processor supports a given feature, then when that processor transforms or assembles information involving that feature, that processor respects the semantics of that feature as described in the Standard.

end note]

This Part of ISO/IEC 29500 defines the following application conformance classes:

- *WML Transitional*, if the application is a conforming application that is a consumer or producer of documents having conformance class WML Transitional.
- *SML Transitional*, if the application is a conforming application that is a consumer or producer of documents having conformance class SML Transitional.
- *PML Transitional*, if the application is a conforming application that is a consumer or producer of documents having conformance class PML Transitional.

3. Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ANSI X3.4-1986, *American Standard Code for Information Interchange (ASCII)*

Bureau of Standards, Metrology and Inspection of the Ministry of Economic Affairs, *CNS 7648: Data Elements and Interchange Formats — Information Interchange — Representation of Dates and Times*

Calendar Reform Committee, Indian Ephemeris and Nautical Almanac. 1957

Stokes, M., M. Anderson, S. Chandrasekar, and R. Motta. *A Standard Default color Space for the Internet. Vers. 1.10*. November 5, 1996. http://www.w3.org/Graphics/Color/sRGB

Har'El, Zvi, *Gauss Formula for the Julian Date of Passover*. Deptartment of Mathematics, Technion, Israel Institue of Technology, Haifa 32000, Israel, 2005, 6

Duerst, M, and M Suignard. *Internationalized Resource Identifiers (IRIs).* IETF. January 2005. http://tools.ietf.org/html/rfc3987

IANA, *Character Sets from IANA*, as specified at http://www.iana.org/assignments/character-sets

IANA. *MIME Media Types*. Internet Assigned Numbers Authority. http://www.iana.org/assignments/media-types/

IEC 60559:1989, *Binary Floating-Point Arithmetic for Microprocessor Systems*

ISO/IEC 2382-1:1993, *Information technology — Vocabulary — Part 1: Fundamental terms*

ISO 8601:2004, *Data elements and interchange formats — Information interchange — Representation of dates and times*

ISO/IEC 8859-1:1998, *Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1* (referred to in ISO/IEC 29500 as the ANSI character set)

ISO/IEC 9075-1, *Information technology — Database languages — SQL — Part 1: Framework (SQL/Framework)*

ISO/IEC 10118-3:2004, *Information technology — Security techniques — Hash-functions — Part 3: Dedicated hash-functions*.

ISO/IEC 10646, *Information technology — Universal Coded Character Set (UCS)*.

ISO/IEC 14496-22:2009, *Information technology — Coding of audio-visual objects — Part 22: Open Font Format*

ISO/IEC 29500-1:2012, *Information technology — Document description and processing languages — Office Open XML File Formats, Part 1: Fundamentals and Markup Language Reference.*

ISO/IEC 29500-3:2011, Information technology — Document description and processing languages — Office Open XML File Formats – Part 3: Markup Compatibility and Extensibility

Japanese Industrial Standard, JIS X 0301: *Data elements and interchange formats —Information interchange — Representation of dates and times*. Japan, 2002.

Kingdom of Saudi Arabia, Ministry of Islamic Affairs, Endowments, Da'wah and Guidance.

Korean Law Enactment No. 4, 1961.

Faure, D. (n.d.). *Creating and Using Components (KParts)*. http://techbase.kde.org/Projects/Documentation.

Maimon, Rabbi Moshe ben, *Complete Restatement of the Oral Law (Mishneh Torah)*.

Ausbrooks, Ron, et al. *Mathematical Markup Language (MathML) Version 2.0 (Second Edition)*. October 21, 2003. http://www.w3.org/TR/MathML/.

Kaliski, B. *The MD2 Message-Digest Algorithm*. April 1992. http://www.ietf.org/rfc/rfc1319.txt

Rivest, R. *The MD4 Message-Digest Algorithm*. April 1992. http://www.ietf.org/rfc/rfc1320.txt

The MD5 Message-Digest Algorithm. April 1992. http://www.ietf.org/rfc/rfc1321.txt

National Measurement Regulations 1999, Commonwealth of Australia http://www.comlaw.gov.au/Details/F2011C00445

NIST Guide to SI Units, http://physics.nist.gov/Pubs/SP811/appenB9.html

QuickTime File Format Specification (2007-09-04 version) http://developer.apple.com/standards/classicquicktime.html

Resource Description Framework (RDF), http://www.w3.org/RDF/

RFC 822, *Standard for ARPA Internet Text Messages* (http://www.ietf.org/rfc/rfc0822.txt)

RFC 2045, Borenstein, N., and N. Freed. *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*. The Internet Society. 1996. http://www.ietf.org/rfc/rfc2045.txt

RFC 2119, Bradner, Scott, 1997: *Key words for use in RFCs to Indicate Requirement Levels*. http://www.ietf.org/rfc/rfc2119.txt

RFC 2616, Berners-Lee, T., R. Fielding, H. Frystyk, J. Gettys, P. Leach, L. Masinter, and J. Mogul. *Hypertext Transfer Protocol—HTTP/1.1*. The Internet Society. 1999. http://www.ietf.org/rfc/rfc2616.txt

RFC 3066, Alvestrand, H. *Tags for the Identification of Languages*. The Internet Society. 2001. http://www.ietf.org/rfc/rfc3066.txt

RFC 3339, Klyne, G. and C. Newman. *Date and Time on the Internet: Timestamps*. The Internet Society. 2002. http://www.ietf.org/rfc/rfc3339.txt

RFC 3629, Yergeau, F. *UTF-8, a transformation format of ISO 10646*. The Internet Society. 2003. http://www.ietf.org/rfc/rfc3629.txt

RFC 3986, Berners-Lee, T., R. Fielding, and L. Masinter. *Uniform Resource Identifier (URI): Generic Syntax*. The Internet Society. 2005. http://www.ietf.org/rfc/rfc3986.txt

Simple Object Access Protocol (SOAP), http://www.w3.org/TR/soap12

SMIL, Bulterman, D., Grassel, G., Jansen, J., Koivisto, A., Layaïda, N., Michel, T., et al. (2005, December 13). *Synchronized Multimedia Integration Language (SMIL 2.1)*. Retrieved from W3C: http://www.w3.org/TR/SMIL/

SVG, Andersson, O., Armstrong, P., Axelsson, H., Berjon, R., Bézaire, B., Bowler, J., et al. (2003, January 14). *Scalable Vector Graphics (SVG) 1.1 Specification*. Retrieved from W3C - World Wide Web Consortium: http://www.w3.org/TR/SVG/

The GNOME Project. (2003, December 12). *Component Model - Bonobo Document Model*. Retrieved from The GNOME Development Site: http://developer.gnome.org/arch/gnome/componentmodel/bonobo.htm

The Unicode Consortium. *The Unicode Standard*, http://www.unicode.org/standard/standard.html

Unicode Technical Report #25, http://www.unicode.org/reports/tr25/

Unicode Technical Note #28, *Nearly Plain-Text Encoding of Mathematics*. August 29, 2006, http://www.unicode.org/notes/tn28

United States Postal Service. *Domestic Mail Manual*. United States Postal Service. November 8, 2007. http://pe.usps.com/cpim/ftp/manuals/dmm300/Full/MailingStandards.pdf

The Units of Measurement Regulations 1995, United Kingdom http://www.opsi.gov.uk/si/si1995/Uksi_19951804_en_2.htm

Universal Postal Union. *POST*CODE: Postal addressing systems*. Berne: UPU Publications, 2006, ISBN 92-95025-37-7, ISSN 1020-6019

Web Accessibility Initiative (WAI), http://www.w3.org/WAI/

XSLT, Clark, James, *XSL Transformations (XSLT) Version 1.0*, World Wide Web Consortium Recommendation. 1999. http://www.w3.org/TR/xslt

XML, Tim Bray, Jean Paoli, Eve Maler, C. M. Sperberg-McQueen, and François Yergeau (editors). *Extensible Markup Language (XML) 1.0*, Fourth Edition.1 World Wide Web Consortium. 2006.

http://www.w3.org/TR/2006/REC-xml-20060816/ [Implementers should be aware that a further correction of the normative reference to XML to refer to the 5th Edition will be necessary when the related Reference Specifications to which this International Standard also makes normative reference and which also depend upon XML, such as XSLT, XML Namespaces and XML Base, are all aligned with the 5th Edition.]

XML Base, Marsh, Jonathan. *XML Base*. World Wide Web Consortium. 2001. http://www.w3.org/TR/2001/REC-xmlbase-20010627/

XML Namespaces, Tim Bray, Dave Hollander, Andrew Layman, and Richard Tobin (editors). *Namespaces in XML 1.0 (Third Edition)*, 8 December 2009. World Wide Web Consortium. http://www.w3.org/TR/2009/REC-xml-names-20091208/

XPATH, Clark, James; DeRose, Steve. *XML Path Language (XPath) Version 1.0*, World Wide Web Consortium Recommendation. 1999. http://www.w3.org/TR/xpath.

XML Schema Part 0: Primer (Second Edition), W3C Recommendation 28 October 2004, http://www.w3.org/TR/xmlschema-0/

XML Schema Part 1: Structures (Second Edition), W3C Recommendation 28 October 2004, http://www.w3.org/TR/xmlschema-1/

XML Schema Part 2: Datatypes (Second Edition), W3C Recommendation 28 October 2004, http://www.w3.org/TR/xmlschema-2/

.ZIP File Format Specification from PKWARE, Inc., version 6.2.0 (2004), as specified in http://www.pkware.com/documents/APPNOTE/APPNOTE_6.2.0.

4. Terms and Definitions

For the purposes of this document, the following terms and definitions apply. Other terms are defined where they appear in *italic* typeface, on the left side of a syntax rule, or within subclauses of language-specific grammars. Terms explicitly defined in this Part of ISO/IEC 29500 are not to be presumed to refer implicitly to similar terms defined elsewhere. [*Note*: This part uses OPC-related terms, which are defined in ISO/IEC 29500-2. *end note*]

application — A consumer or producer.

behavior — External appearance or action.

behavior, implementation-defined — Unspecified behavior where each implementation is expected to document that behavior, which would thereby promote predictability and reproducibility within any given implementation. (This term is sometimes called “application-defined behavior”.)

behavior, locale-specific — Behavior that depends on local conventions of nationality, culture, and language.

behavior, unspecified —Behavior where ISO/IEC 29500 makes no recommendations. [*Note*: To add an extension, an implementer must use the extensibility mechanisms described by ISO/IEC 29500 rather than trying to do so by giving meaning to otherwise unspecified behavior. *end note*]

byte — A sequence of 8 bits treated as a unit.

comment — A note that an author or reviewer attaches to content in a document. Although a consumer might choose to display comments, they are not considered part of the body of the document. A comment might include the text of the note, the comment author's name and initials, and date of creation, among other things.

consumer — A piece of software or a device that reads packages through a package implementer. A consumer is often designed to consume packages only for a specific physical package format.

content type — Describes the content stored in a part. Content types define a media type, a subtype, and an optional set of parameters, as defined in RFC 2616.

document category — One of the three categories of Office Open XML documents: Wordprocessing, Spreadsheet, and Presentation, defined as follows:

- A document whose package-relationship item contains a relationship to a Main Document part (Part 1, §11.3.10) is a document of category Wordprocessing.
- A document whose package-relationship item contains a relationship to a Workbook part (Part 1, §12.3.23) is a document of category Spreadsheet.

- A document whose package-relationship item contains a relationship to a Presentation part (Part 1, §13.3.6) is a document of category Presentation.

An Office Open XML document can contain one or more embedded Office Open XML packages (Part 1, §15.2.11) with each embedded package having any of the three document categories. However, the presence of these embedded packages does not change the category of the document.

DOS file path —A legacy file naming scheme which used a file name of at most eight characters, followed by a period ("."), followed by a filename extension of at most three characters. This name may be preceded by a slash-delimited path, and the combined structure may be preceded by a drive letter specifier. The grammar for DOS file paths is defined as follows:

```
filepath = [drive] [ folder] [filename]
filename = corefilename "." (0 * 3 validchar)
corefilename = (1 * 8 validchar)
validchar = uppercaseletter | decimaldigit | "!" | "#" | "$" | "%" | "&"
  | "'" | "\|" | "-" | "@" | "^" | "_" | "`" | "{" | "}" | "~"
corefoldername = (1 * 8 validchar) | "." | ".."
folder = 1 * (corefoldername "\\") | "\\"
drive = uppercaseletter ":\\"
uppercaseletter = "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J"
  | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V"
  | "W" | "X" | "Y" | "Z"
decimaldigit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
```

DrawingML — A set of conventions for specifying the location and appearance of drawing elements in an Office Open XML document.

extension — Any XML element, XML attribute, relationship, or part not explicitly included in ISO/IEC 29500, but that uses the extensibility mechanisms described by ISO/IEC 29500.

Office Open XML document — A rendition of a data stream formatted using the wordprocessing, spreadsheet, or presentation ML and its related MLs as described in ISO/IEC 29500-1 and ISO/IEC 29500-4. Such a document is represented as a package as described in ISO/IEC 29500-2.

package— A ZIP archive that conforms to the Open Packaging Conventions specification defined in ISO/IEC 29500-2.

package, embedded— A package that has been stored as the target of an Embedded Package relationship (Part 1, §15.2.11) in an Office Open XML document

PresentationML — A set of conventions for representing an Office Open XML document of category Presentation.

producer — A piece of software or a device that writes packages through a package implementer. A producer is often designed to produce packages according to a particular physical package format specification.

relationship —The kind of connection between a source part and a target part in a package. Relationships make the connections between parts directly discoverable without looking at the content in the parts, and without altering the parts themselves. (See also Package Relationships.)

relationships part — A part containing an XML representation of relationships.

relationship, explicit — A relationship in which a resource is referenced from a source part's XML using the Id attribute of a Relationship tag.

relationship, implicit — A relationship that is not explicit.

SpreadsheetML — A set of conventions for representing an Office Open XML document of category Spreadsheet.

WordprocessingML — A set of conventions for representing an Office Open XML document of category Wordprocessing.

5. Notational Conventions

The following typographical conventions are used in this Part of ISO/IEC 29500:

1. The first occurrence of a new term is written in italics. [*Example*: The text in ISO/IEC 29500 is divided into *normative* and *informative* categories. *end example*]
2. In each definition of a term in §4 (Terms and Definitions), the term is written in bold. [*Example*: **behavior** — External appearance or action. *end example*]
3. The tag name of an XML element is written using a distinct style and typeface. [*Example*: The bookmarkStart and bookmarkEnd elements specify … *end example*]
4. The name of an XML attribute is written using a distinct style and typeface. [*Example*: The dropCap attribute specifies … *end example*]
5. The value of an XML attribute is written using a constant-width style. [*Example*: The attribute value of `auto` specifies … *end example*]
6. The qualified or unqualified name of a simple type, complex type, or base datatype is written using a distinct style and typeface. [*Example*: The possible values for this attribute are defined by the ST_HexColor simple type. *end example*]

6. Acronyms and Abbreviations

This clause is informative.

The following acronyms and abbreviations are used throughout ISO/IEC 29500:

IEC — the International Electrotechnical Commission

ISO — the International Organization for Standardization

W3C — World Wide Web Consortium

End of informative text.

7. General Description

This Part of ISO/IEC 29500 is divided into the following subdivisions:

1. Front matter (clauses 1–7);
2. Main body (clauses 8–20);
3. Annexes

Examples are provided to illustrate possible forms of the constructions described. References are used to refer to related clauses. Notes are provided to give advice or guidance to implementers or programmers. Rationale provides explanatory material as to why something is or is not in ISO/IEC 29500. Annexes provide additional information or summarize the information contained in ISO/IEC 29500.

The following form the normative pieces of this Part of ISO/IEC 29500:

- Introduction
- Clauses 1–5, 7, and 8–20
- Annex A

The following form the informative pieces of this Part of ISO/IEC 29500:

- Clause 6
- Annex B–Annex D
- All notes
- All examples

Except for whole clauses or annexes that are identified as being informative, informative text that is contained within normative text is indicated in the following ways:

1. [*Example:* code fragment, possibly with some narrative … *end example*]
2. [*Note:* narrative … *end note*]
3. [*Rationale:* narrative … *end rationale*]
4. [*Guidance*: narrative … *end guidance*]

Unless stated otherwise in this Part, the functionality defined in Part 1 is applicable to Part 4. However, Part 4 uses namespaces that are different from those used by Part 1. As such, when examples in Part 1 are read in the context of Part 4, they should be understood in the context of the corresponding Part 4 namespaces.

8. Additional Shared Parts

8.1 VML Drawing Part

Content Type:	application/vnd.openxmlformats-officedocument.vmlDrawing
Root Namespace:	not applicable
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/vmlDrawing

An instance of this part type contains markup in the Vector Markup Language (VML) syntax, which is used to provide an alternative image representation of objects stored in a SpreadsheetML or PresentationML document.

[*Note*: The VML format is a legacy format originally introduced with Office 2000 and is included and fully defined in ISO/IEC 29500 for backwards compatibility reasons. The DrawingML format is a newer and richer format created with the goal of eventually replacing any uses of VML in the Office Open XML formats. VML should be considered a deprecated format included in Office Open XML for legacy reasons only and new applications that need a file format for drawings are strongly encouraged to use preferentially DrawingML. *end note*]

A package is permitted to contain zero or more VML Drawing parts, each of which shall be the target of an explicit relationship in a Handout Master (Part 1, §13.3.3), Notes Slide (Part 1, §13.3.5), Notes Master (Part 1, §13.3.4), Slide (Part 1, §13.3.8), Slide Layout (Part 1, §13.3.9), or Slide Master (Part 1, §13.3.10) part in a PresentationML document; or a Dialogsheet (Part 1, §12.3.7) or Worksheet part (Part 1, §12.3.24) in a SpreadsheetML document.

[*Example*: The following SpreadsheetML's package-relationship item contains one relationship, for the VML Drawing part stored in the ZIP item ../drawings/drawing1.vml:

```
<Relationships xmlns="…">
  <Relationship Id="rId8"
    Type="http://…/vmlDrawing" Target="../drawings/drawing1.vml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be xml in the null namespace, encapsulating an arbitrary amount of VML markup as defined by ISO/IEC 29500.

[*Example*: Consider the following VML Drawing part:

```
<xml>
  <v:shape ...>
    ...
  </v:shape>
  ...
</xml>
```

end example]

A VML Drawing part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A VML Drawing part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

- Image (Part 1, §15.2.14)

A VML Drawing part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

9. WordprocessingML

The following parts, which are defined in subclauses within Part 1, §11, "WordprocessingML", have different source relationships and/or root namespaces when used in documents of the Transitional conformance class:

9.1 Part Summary (Part 1, §11.3)

9.1.1 Alternative Format Import Part (Part 1, §11.3.1)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/aFChunk

9.1.2 Comments Part (Part 1, §11.3.2)

Root Namespace:	http://schemas.openxmlformats.org/wordprocessingml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/comments

9.1.3 Document Settings Part (Part 1, §11.3.3)

Root Namespace:	http://schemas.openxmlformats.org/wordprocessingml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/settings

9.1.4 Endnotes Part (Part 1, §11.3.4)

Root Namespace:	http://schemas.openxmlformats.org/wordprocessingml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/endnotes

9.1.5 Fonts Table Part (Part 1, §11.3.5)

Root Namespace:	http://schemas.openxmlformats.org/wordprocessingml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/fontTable

9.1.6 Footer Part (Part 1, §11.3.6)

Root Namespace:	http://schemas.openxmlformats.org/wordprocessingml/2006/main

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer

9.1.7 Footnotes Part (Part 1, §11.3.7)

Root Namespace:	http://schemas.openxmlformats.org/wordprocessingml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/footnotes

9.1.8 Glossary Document Part (Part 1, §11.3.8)

Root Namespace:	http://schemas.openxmlformats.org/wordprocessingml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/glossaryDocument

9.1.9 Header Part (Part 1, §11.3.9)

Root Namespace:	http://schemas.openxmlformats.org/wordprocessingml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/header

9.1.10 Main Document Part (Part 1, §11.3.10)

Root Namespace:	http://schemas.openxmlformats.org/wordprocessingml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument

9.1.11 Numbering Definitions Part (Part 1, §11.3.11)

Root Namespace:	http://schemas.openxmlformats.org/wordprocessingml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/numbering

9.1.12 Style Definitions Part (Part 1, §11.3.12)

Root Namespace:	http://schemas.openxmlformats.org/wordprocessingml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/styles

9.1.13 Web Settings Part (Part 1, §11.3.13)

Root	http://schemas.openxmlformats.org/wordprocessingml/2006/main

Namespace:	
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/webSettings

9.2 Document Template (Part 1, §11.4)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/attachedTemplate

9.3 Framesets (Part 1, §11.5)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/frame

9.4 Master Documents and Subdocuments (Part 1, §11.6)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/subDocument

9.5 Mail Merge Data Source (Part 1, §11.7)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/mailMergeSource

9.6 Mail Merger Header Data Source (Part 1, §11.8)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/mailMergeHeaderSource

9.7 XSL Transformation (Part 1, §11.9)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/transform

10. SpreadsheetML

The following parts, which are defined in subclauses within Part 1, §12, "SpreadsheetML", have different source relationships and/or root namespaces when used in documents of the Transitional conformance class:

10.1 Part Summary (Part 1, §12.3)

10.1.1 Calculation Chain Part (Part 1, §12.3.1)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/calcChain

10.1.2 Chartsheet Part (Part 1, §12.3.2)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/chartsheet

10.1.3 Comments Part (Part 1, §12.3.3)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/comments

10.1.4 Connections Part (Part 1, §12.3.4)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/connections

10.1.5 Custom Property Part (Part 1, §12.3.5)

Root Namespace:	Not applicable
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/customProperty

10.1.6 Custom XML Mappings Part (Part 1, §12.3.6)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/xmlMaps

10.1.7 Dialogsheet Part (Part 1, §12.3.7)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/dialogsheet

10.1.8 Drawings Part (Part 1, §12.3.8)

Root Namespace:	http://schemas.openxmlformats.org/drawingml/2006/spreadsheetDrawing
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/drawing

10.1.9 External Workbook References Part (Part 1, §12.3.9)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/externalLink

10.1.10 Metadata Part (Part 1, §12.3.10)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/sheetMetadata

10.1.11 Pivot Table Part (Part 1, §12.3.11)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/pivotTable

10.1.12 Pivot Table Cache Definition Part (Part 1, §12.3.12)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/pivotCacheDefinition

10.1.13 Pivot Table Cache Records Part (Part 1, §12.3.13)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/pivotCacheRecords

10.1.14 Query Table Part (Part 1, §12.3.14)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/queryTable

10.1.15 Shared Strings Table Part (Part 1, §12.3.15)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/sharedStrings

10.1.16 Shared Workbook Revision Headers Part (Part 1, §12.3.16)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/revisionHeaders

10.1.17 Shared Workbook Revision Log Part (Part 1, §12.3.17)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/revisionLog

10.1.18 Shared Workbook User Data part (Part 1, §12.3.18)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/usernames

10.1.19 Single Cell Table Definitions Part (Part 1, §12.3.19)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/tableSingleCells

10.1.20 Styles Part (Part 1, §12.3.20)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/styles

10.1.21 Table Definition Part (Part 1, §12.3.21)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/table

10.1.22 Volatile Dependencies Part (Part 1, §12.3.22)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/volatileDependencies

10.1.23 Workbook Part (Part 1, §12.3.23)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument

10.1.24 Worksheet Part (Part 1, §12.3.24)

Root Namespace:	http://schemas.openxmlformats.org/spreadsheetml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/worksheet

10.2 External Workbooks (Part 1, §12.4)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/externalLinkPath

11. PresentationML

The following parts, which are defined in subclauses within Part 1, §13, "PresentationML", have different source relationships and/or root namespaces when used in documents of the Transitional conformance class:

11.1 Part Summary (Part 1, §13.3)

11.1.1 Comment Authors Part (Part 1, §13.3.1)

Root Namespace:	http://schemas.openxmlformats.org/presentationml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/commentAuthors

11.1.2 Comments Part (Part 1, §13.3.2)

Root Namespace:	http://schemas.openxmlformats.org/presentationml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/comments

11.1.3 Handout Master Part (Part 1, §13.3.3)

Root Namespace:	http://schemas.openxmlformats.org/presentationml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/handoutMaster

11.1.4 Notes Master Part (Part 1, §13.3.4)

Root Namespace:	http://schemas.openxmlformats.org/presentationml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/notesMaster

11.1.5 Notes Slide Part (Part 1, §13.3.5)

Root Namespace:	http://schemas.openxmlformats.org/presentationml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/notesSlide

11.1.6 Presentation Part (Part 1, §13.3.6)

Root Namespace:	http://schemas.openxmlformats.org/presentationml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument

11.1.7 Presentation Properties Part (Part 1, §13.3.7)

Root Namespace:	http://schemas.openxmlformats.org/presentationml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/presProps

11.1.8 Slide Part (Part 1, §13.3.8)

Root Namespace:	http://schemas.openxmlformats.org/presentationml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/slide

11.1.9 Slide Layout Part (Part 1, §13.3.9)

Root Namespace:	http://schemas.openxmlformats.org/presentationml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/slideLayout

11.1.10 Slide Master Part (Part 1, §13.3.10)

Root Namespace:	http://schemas.openxmlformats.org/presentationml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/slideMaster

11.1.11 Slide Synchronization Data Part (Part 1, §13.3.11)

Root Namespace:	http://schemas.openxmlformats.org/presentationml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/slideUpdateInfo

11.1.12 User Defined Tags Part (Part 1, §13.3.12)

Root Namespace:	http://schemas.openxmlformats.org/presentationml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/tags

11.1.13 View Properties Part (Part 1, §13.3.13)

Root Namespace:	http://schemas.openxmlformats.org/presentationml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/viewProps

11.2 HTML Publish Location (Part 1, §13.4)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/ htmlPubSaveAs

11.3 Slide Synchronization Server Location (Part 1, §13.5)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/slideUpdateUrl

12. DrawingML

The following parts, which are defined in subclauses within Part 1, §14, "DrawingML", have different source relationships and/or root namespaces when used in documents of the Transitional conformance class:

12.1 Part Summary (Part 1, §14.2)

12.1.1 Chart Part (Part 1, §14.2.1)

Root Namespace:	http://schemas.openxmlformats.org/drawingml/2006/chart
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/chart

12.1.2 Chart Drawing Part (Part 1, §14.2.2)

Root Namespace:	http://schemas.openxmlformats.org/drawingml/2006/chart
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/chartUserShapes

12.1.3 Diagram Colors Part (Part 1, §14.2.3)

Root Namespace:	http://schemas.openxmlformats.org/drawingml/2006/diagram
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/diagramColors

12.1.4 Diagram Data Part (Part 1, §14.2.4)

Root Namespace:	http://schemas.openxmlformats.org/drawingml/2006/diagram
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/diagramData

12.1.5 Diagram Layout Definition Part (Part 1, §14.2.5)

Root Namespace:	http://schemas.openxmlformats.org/drawingml/2006/diagram
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/diagramLayout

12.1.6 Diagram Style Part (Part 1, §14.2.6)

Root Namespace:	http://schemas.openxmlformats.org/drawingml/2006/diagram
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/diagramQuickStyle

12.1.7 Theme Part (Part 1, §14.2.7)

Root Namespace:	http://schemas.openxmlformats.org/drawingml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/theme

12.1.8 Theme Override Part (Part 1, §14.2.8)

Root Namespace:	http://schemas.openxmlformats.org/drawingml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/themeOverride

12.1.9 Table Styles Part (Part 1, §14.2.9)

Root Namespace:	http://schemas.openxmlformats.org/drawingml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/tableStyles

13. Shared MLs

The following parts, which are defined in subclauses within Part 1, §15, "Shared", have different source relationships and/or root namespaces when used in documents of the Transitional conformance class:

13.1 Part Summary (Part 1, §15.2)

13.1.1 Additional Characteristics Part (Part 1, §15.2.1)

Root Namespace:	http://schemas.openxmlformats.org/officeDocument/2006/additionalCharacteristics
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/customXml

13.1.2 Audio Part (Part 1, §15.2.2)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/audio

13.1.3 Bibliography Part (Part 1, §15.2.3)

Root Namespace:	http://schemas.openxmlformats.org/officeDocument/2006/bibliography
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/customXml

13.1.4 Content Part (Part 1, §15.2.4)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/customXml

13.1.5 Custom XML Data Storage Part (Part 1, §15.2.5)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/customXml

13.1.6 Custom XML Data Storage Properties Part (Part 1, §15.2.6)

Root Namespace:	http://schemas.openxmlformats.org/officeDocument/2006/customXmlDataProps
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/customXmlProps

13.1.7 Embedded Control Persistence Part (Part 1, §15.2.9)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/control

13.1.8 Embedded Object Part (Part 1, §15.2.10)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/oleObject

13.1.9 Embedded Package Part (Part 1, §15.2.11)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/package

13.1.10 Core File Properties Part (Part 1, §15.2.12.1)

Source Relationship:	http://schemas.openxmlformats.org/officedocument/2006/relationships/metadata/core-properties

13.1.11 Custom File Properties Part (Part 1, §15.2.12.2)

Root Namespace:	http://schemas.openxmlformats.org/officeDocument/2006/custom-properties
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/custom-properties

13.1.12 Extended File Properties Part (Part 1, §15.2.12.3)

Root Namespace:	http://schemas.openxmlformats.org/officeDocument/2006/extended-properties
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/extended-properties

13.1.13 Font Part (Part 1, §15.2.13)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/font

13.1.14 Image Part (Part 1, §15.2.14)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/image

13.1.15 Printer Settings Part (Part 1, §15.2.15)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings

13.1.16 Thumbnail Part (Part 1, §15.2.16)

Source Relationship:	http://schemas.openxmlformats.org/package/2006/relationships/metadata/thumbnail

13.1.17 Video Part (Part 1, §15.2.17)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/video

13.2 Hyperlinks Part (Part 1, §15.3)

Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink

14. WordprocessingML Reference Material

[*Note*: For further information on the mapping of elements and attributes to OPC parts, see the Bibliography entry, "Information on elements, attributes, and OPC parts in ISO/IEC 29500 (OOXML)". *end note*]

14.1 Table of Contents

This subclause is informative.

End of informative text.

14.2 Paragraphs and Rich Formatting

14.2.1 Paragraphs

14.2.1.1 Additional attribute for cnfStyle element (Part 1, §17.3.1.8)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description
val (Conditional Formatting Bit Mask)	Specifies the set of conditional formatting properties that have been applied to this object. These properties are expressed using a string serialization of a binary bitmask for each of the following properties (reading from the first character position right): • First Row - Is this the first row of the table? • Last Row - Is this the last row of the table? • First Column - Does this belong to the first column of the table? • Last Column - Does this belong to the last column of the table? • Band 1 Vertical - Does this belong to a column which should receive band 1 formatting? This property specifies whether the cell should receive the formatting specified for odd-numbered columns (e.g. 1,3,5,…) • Band 2 Vertical - Does this belong to a column which should receive band 2 formatting? This property specifies whether the cell should receive the formatting specified for even-numbered columns (e.g. 2,4,6…) • Band 1 Horizontal - Does this receive band 1 formatting? This property specifies whether the cell should receive the formatting specified for odd-numbered rows (e.g. 1,3,5,…) • Band 2 Horizontal - Does this receive band 2 formatting? This property specifies whether the cell should receive the formatting specified for even-numbered rows (e.g. 2,4,6…) • NE Cell - Is this part of the top-right corner of the table?

Attributes	Description

	• NW Cell - Is this part of the top-left corner of the table?
	• SE Cell - Is this part of the bottom-right corner of the table?
	• SW Cell - Is this part of the bottom-left corner of the table?
	
	For each of these properties, a value of 1 in the specified character position in the string means that the value is true, a value of 0 means false. All values shall be specified.
	
	[*Example*: Consider a paragraph in the top right corner of a table with a table style applied. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:val="101000000100" />`
	`  …`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph specifies that it has the conditional properties from the table style for the first column, first row, and the NW corner of the parent table by setting the appropriate bits in the val attribute. *end example*]
	
	The possible values for this attribute are defined by the ST_Cnf simple type (§14.10.8).

14.2.1.2 Additional attributes for ind element (Part 1, §17.3.1.12)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description
left (Start Indentation)	Semantically equivalent to the start attribute. The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (Part 1, §17.18.81).
leftChars (Start Indentation in Character Units)	Semantically equivalent to the startChars attribute. The possible values for this attribute are defined by the ST_DecimalNumber simple type (Part 1, §17.18.10).
right (End Indentation)	Semantically equivalent to the end attribute. The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (Part 1, §17.18.81).
rightChars (End Indentation in Character Units)	Semantically equivalent to the endChars attribute. The possible values for this attribute are defined by the ST_DecimalNumber simple type (Part 1, §17.18.10).

14.2.2 Run Content

14.2.2.1 control (Floating Embedded Control)

This element specifies that the parent VML object is a representation of an embedded control at the current location in the document. This element shall be used to associate the VML data with the appropriate embedded control settings and properties when the document is displayed.

If the embedded control is not present, cannot be loaded due to application settings, or is not supported, then the VML data shall be used to provide an image representation of the control at the appropriate location in the document.

[*Example*: Consider a run which consists of an embedded control. That run would be specified using the following WordprocessingML:

```
<w:r>
  <w:pict>
    …
    <w:control r:id="rId99" w:shapeid="shape01" … />
  </w:pict>
</w:r>
```

The control element indicates that the parent VML object contains the positioning and last known image representation of an embedded control, whose settings and properties are stored on this element. *end example*]

Attributes	Description
id (Embedded Control Properties Relationship Reference) Namespace: …/officeDocument/2006/relationships	Specifies the relationship ID for the relationship which contains the properties for this embedded control. This property bag is contained in a separate part within the Office Open XML package. The relationship explicitly targeted by this attribute shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/control or the document shall be conisdered non-conformant. If this attribute is omitted, then the embedded control shall be given no property bag when instantiated. [*Example*: Consider the following WordprocessingML markup for an embedded control in a document: `<w:control r:id="rId5" w:name="CheckBox1" w:shapeid="_x0000_s1027" />` The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rId5` must contain the property data for this embedded control. *end example*]

Attributes	Description
	The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).
name (Unique Name for Embedded Control)	Specifies a unique name for this embedded control. This name shall be unique across all controls in this document. [*Example*: Consider the following WordprocessingML markup for an embedded control in a document: `<w:control r:id="rId5" w:name="CheckBox1" w:shapeid="_x0000_s1027" />` The name attribute specifies that the unique name for this control must be `CheckBox1`. *end example*] The possible values for this attribute are defined by the ST_String simple type (Part 1, §22.9.2.13).
shapeid (Shape Reference)	Specifies the shape ID for a shape which shall be used to define the presentation and location of this embedded control within the document if the control is floating using the DrawingML syntax. [*Note*: This positioning data is sufficient to display the control in any case where: • The embedded control is not on the current machine • Embedded controls are disabled • Embedded controls of this control type are not supported *end note*] This shape ID reference is resolved by looking for a DrawingML object whose id attribute matches the value specified within this attribute. If no such shape exists, then the control shall be rendered inline in the document content at the current run content location. If this attribute is omitted, then this embedded control shall be displayed inline in the current location in the parent run. [*Example*: Consider the following WordprocessingML markup for an embedded control in a document: `<w:control r:id="rId5" w:name="CheckBox1" w:shapeid="10" />` The shapeid attribute specifies that the DrawingML object with an id attribute value of `10` must contain the positioning data for this embedded control. *end example*] The possible values for this attribute are defined by the ST_String simple type (Part 1, §22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Control) is located in §A.1. *end note*]

14.2.2.2 pict (VML Object)

This element specifies that an object is located at this position in the run's contents. The layout properties of this object are specified using the VML syntax (§19.1).

[*Example*: Consider a run which consists of an object specified using VML. That run would be specified using the following WordprocessingML:

```
<w:r>
  <w:pict>
    …
  </w:pict>
</w:r>
```

The pict element indicates that an object specified in VML is located at the current position in the run (e.g. a floating embedded control). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Picture) is located in §A.1. *end note*]

14.3 Tables

14.3.1 left (Table Cell Leading Edge Border)

This element is semantically equivalent to start (Part 1, §17.4.34), specified above.

For tables which have the bidiVisual property (Part 1, §17.4.1) applied, this border is applied to the right edge of the cell.

This element's content model is defined by the common border properties definition in Part 1, §17.3.4.

14.3.2 left (Table Leading Edge Border)

This element is semantically equivalent to start (Part 1, §17.4.37), specified above.

For tables which have the bidiVisual property (Part 1, §17.4.1) applied, this border is applied to the right edge of the table.

This element's content model is defined by the common border properties definition in Part 1, §17.3.4.

14.3.3 left (Table Cell Leading Margin Exception)

This element is semantically equivalent to start (Part 1, §17.4.36), specified above.

For tables which have the bidiVisual property (Part 1, §17.4.1) applied, this cell margin is applied to the right edge of the cell.

This element's content model is defined by the common table measurement definition in Part 1, §17.4.88.

14.3.4 left (Table Cell Leading Margin Default)

This element is semantically equivalent to start (Part 1, §17.4.35), specified above.

For tables which have the bidiVisual property (Part 1, §17.4.1) applied, this cell margin is applied to the right edge of the cell.

This element's content model is defined by the common table measurement definition in Part 1, §17.4.88.

14.3.5 right (Table Cell Trailing Edge Border)

This element is semantically equivalent to end (Part 1, §17.4.12), specified above.

For tables which have the bidiVisual property (Part 1, §17.4.1) applied, this border is applied to the left edge of the cell.

This element's content model is defined by the common border properties definition in Part 1, §17.3.4.

14.3.6 right (Table Trailing Edge Border)

This element is semantically equivalent to end (Part 1, §17.4.13), specified above.

For tables which have the bidiVisual property (Part 1, §17.4.1) applied, this border is applied to the left edge of the table.

This element's content model is defined by the common border properties definition in Part 1, §17.3.4.

14.3.7 right (Table Cell Trailing Margin Default)

This element is semantically equivalent to end (Part 1, §17.4.11), specified above.

For tables which have the bidiVisual property (Part 1, §17.4.1) applied, this cell margin is applied to the left edge of the cell.

This element's content model is defined by the common table measurement definition in Part 1, §17.4.88.

14.3.8 right (Table Cell Trailing Margin Exception)

This element is semantically equivalent to end (Part 1, §17.4.10), specified above.

For tables which have the bidiVisual property (Part 1, §17.4.1) applied, this cell margin is applied to the left edge of the cell.

This element's content model is defined by the common table measurement definition in Part 1, §17.4.88.

14.3.9 Additional attribute for cnfStyle element (Part 1, §17.4.7)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description

val (Conditional Formatting Bit Mask)	Specifies the set of conditional formatting properties that have been applied to this object.
	
	These properties are expressed using a string serialization of a binary bitmask for each of the following properties (reading from the first character position right):
	
	• First Row - Is this the first row of the table?
	• Last Row - Is this the last row of the table?
	• First Column - Does this belong to the first column of the table?
	• Last Column - Does this belong to the last column of the table?
	• Band 1 Vertical - Does this belong to a column which should receive band 1 formatting? This property specifies whether the cell should receive the formatting specified for odd-numbered columns (e.g. 1,3,5,…)
	• Band 2 Vertical - Does this belong to a column which should receive band 2 formatting? This property specifies whether the cell should receive the formatting specified for even-numbered columns (e.g. 2,4,6…)
	• Band 1 Horizontal - Does this receive band 1 formatting? This property specifies whether the cell should receive the formatting specified for odd-numbered rows (e.g. 1,3,5,…)
	• Band 2 Horizontal - Does this receive band 2 formatting? This property specifies whether the cell should receive the formatting specified for even-numbered rows (e.g. 2,4,6…)
	• NE Cell - Is this part of the top-right corner of the table?
	• NW Cell - Is this part of the top-left corner of the table?
	• SE Cell - Is this part of the bottom-right corner of the table?
	• SW Cell - Is this part of the bottom-left corner of the table?
	
	For each of these properties, a value of `1` in the specified character position in the string means that the value is true, a value of `0` means false. All values shall be specified.
	
	[*Example*: Consider a paragraph in the top right corner of a table with a table style applied. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:val="101000000100" />`
	`  …`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph specifies that it has the conditional properties from the table style for the first column, first row, and the NW corner of the parent table by setting the appropriate bits in the val attribute. *end example*]
	
	The possible values for this attribute are defined by the ST_Cnf simple type (§14.10.8).

14.3.10 Additional attribute for cnfStyle element (Part 1, §17.4.8)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description

val (Conditional Formatting Bit Mask)	Specifies the set of conditional formatting properties that have been applied to this object.
	
	These properties are expressed using a string serialization of a binary bitmask for each of the following properties (reading from the first character position right):
	
	• First Row - Is this the first row of the table?
	• Last Row - Is this the last row of the table?
	• First Column - Does this belong to the first column of the table?
	• Last Column - Does this belong to the last column of the table?
	• Band 1 Vertical - Does this belong to a column which should receive band 1 formatting? This property specifies whether the cell should receive the formatting specified for odd-numbered columns (e.g. 1,3,5,…)
	• Band 2 Vertical - Does this belong to a column which should receive band 2 formatting? This property specifies whether the cell should receive the formatting specified for even-numbered columns (e.g. 2,4,6…)
	• Band 1 Horizontal - Does this receive band 1 formatting? This property specifies whether the cell should receive the formatting specified for odd-numbered rows (e.g. 1,3,5,…)
	• Band 2 Horizontal - Does this receive band 2 formatting? This property specifies whether the cell should receive the formatting specified for even-numbered rows (e.g. 2,4,6…)
	• NE Cell - Is this part of the top-right corner of the table?
	• NW Cell - Is this part of the top-left corner of the table?
	• SE Cell - Is this part of the bottom-right corner of the table?
	• SW Cell - Is this part of the bottom-left corner of the table?
	
	For each of these properties, a value of 1 in the specified character position in the string means that the value is true, a value of 0 means false. All values shall be specified.
	
	[*Example*: Consider a paragraph in the top right corner of a table with a table style applied. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:val="101000000100" />`
	`  …`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph specifies that it has the conditional properties from the table style for the

Attributes	Description
	first column, first row, and the NW corner of the parent table by setting the appropriate bits in the val attribute. *end example*] The possible values for this attribute are defined by the ST_Cnf simple type (§14.10.8).

14.3.11 Additional attribute for tblLook element (Part 1, §17.4.55)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description

val (Bitmask of Table Conditional Formatting)	Specifies a hexadecimal code containing a bitmask of options, interpreted as follows:
	• 0x0020=Apply first row conditional formatting
	• 0x0040=Apply last row conditional formatting
	• 0x0080=Apply first column conditional formatting
	• 0x0100=Apply last column conditional formatting
	• 0x0200=Do not apply row banding conditional formatting
	• 0x0400=Do not apply column banding conditional formatting
	
	If omitted, the bitmask of table style options on the current table shall be assumed to be `0000`.
	
	[*Example*: Consider a table which must use the following conditional formatting properties from the referenced table style:
	• First row conditional formatting
	• Last row conditional formatting
	
	This table would then apply the following portions of the bitmask:
	• 0x0020=Apply first row conditional formatting
	• 0x0040=Apply last row conditional formatting
	• 0x0200=Do not apply row banding conditional formatting
	• 0x0400=Do not apply column banding conditional formatting
	
	The resulting WordprocessingML would be specified as follows:
	
	`<w:tblPr>`
	`  <w:tblLook w:val="0660"/>`
	`</w:tblPr>`
	
	The val attribute specifies a bitmask which determines the components of the table style applied to the current table. *end example*]
	
	The possible values for this attribute are defined by the ST_ShortHexNumber simple type (Part 1, §17.18.79).

14.3.12 Additional attribute for tblLook element (Part 1, §17.4.56)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description

val (Bitmask of Table Conditional Formatting)	Specifies a hexadecimal code containing a bitmask of options, interpreted as follows:
	• 0x0020=Apply first row conditional formatting
	• 0x0040=Apply last row conditional formatting
	• 0x0080=Apply first column conditional formatting
	• 0x0100=Apply last column conditional formatting
	• 0x0200=Do not apply row banding conditional formatting
	• 0x0400=Do not apply column banding conditional formatting
	
	If omitted, the bitmask of table style options on the current table shall be assumed to be `0000`.
	
	[*Example*: Consider a table which must use the following conditional formatting properties from the referenced table style:
	• First row conditional formatting
	• Last row conditional formatting
	
	This table would then apply the following portions of the bitmask:
	• 0x0020=Apply first row conditional formatting
	• 0x0040=Apply last row conditional formatting
	• 0x0200=Do not apply row banding conditional formatting
	• 0x0400=Do not apply column banding conditional formatting
	
	The resulting WordprocessingML would be specified as follows:
	
	`<w:tblPr>`
	`  <w:tblLook w:val="0660"/>`
	`</w:tblPr>`
	
	The val attribute specifies a bitmask which determines the components of the table style applied to the current table. *end example*]
	
	The possible values for this attribute are defined by the ST_ShortHexNumber simple type (Part 1, §17.18.79).

14.3.13 hMerge (Horizontally Merged Cell)

This element specifies that this cell is part of a horizontally merged set of cells in a table. The val attribute on this element determines how this cell is defined with respect to the previous cell in the table (i.e., whether this cell continues the horizontal merge or starts a new merged group of cells).

[*Note*: This property is maintained for compatibility with legacy word processing documents that defined tables in this manner. Whenever possible, this form or horizontal merges should not be produced, and should be translated to the appropriate gridSpan (Part 1) settings on the table cells instead. *end note*]

If this element is omitted, then this cell shall not be part of any horizontally merged grouping of cells, and any horizontal merge group in the preceding cells shall be closed.

[*Example*: Consider a table with one row and three columns with the last two columns horizontally merged:

The second cell in the first row starts a merge that is completed in the right adjacent cell, resulting in the following WordprocessingML:

```
<w:tbl>
   …
  <w:tr>
     <w:tc>
       …
    </w:tc>
     <w:tc>
       <w:tcPr>
         <w:hMerge w:val="restart"/>
      </w:tcPr>
      …
    </w:tc>
    <w:tc>
      <w:tcPr>
        <w:hMerge/>
      </w:tcPr>
      …
    </w:tc>
  </w:tr>
</w:tbl>
```

The hMerge element defines the cells that are to be horizontally merged, and how each group is merged together. *end example*]

Attributes	Description

val (Horizontal Merge Type)	Specifies how the table cell is part of a horizontally merged region. This determines whether the cell should join onto an existing grouping of merged cells if any exist, or start a new group of merged cells. Refer to the simple type definition for a full description of each type.
	
	If this attribute is omitted, its value shall be assumed to be continue.
	
	[*Example*: Consider a table cell where a horizontal cell merge begins represented as the following WordprocessingML:
	
	`<w:tcPr>`
	`  <w:hMerge w:val="restart"/>`
	`</w:tcPr>`
	
	The attribute value of restart specifies that this element must start a new horizontally merged region in this table. *end example*]
	
	The possible values for this attribute are defined by the ST_Merge simple type (Part 1, §17.18.57).

[*Note*: The W3C XML Schema definition of this element's content model (CT_HMerge) is located in §A.1. *end note*]

14.4 Fonts

14.4.1 Elements

14.4.1.1 Additional attribute for charset element (Part 1, §17.8.3.2)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description
val (Value)	Specifies a value specified as single octet (two-digit) hexadecimal number whose contents are interpreted based on the context of the parent XML element. If this attribute is not present, then the character set for this font shall be assumed to be ISO/IEC 8859-1. The value of this attribute shall be interpreted as follows: **Value** — **Description** `0x00` — Specifies a Latin character set. (IANA name `iso-8859-1`) `0x01` — Specifies the default character set. `0x02` — Specifies the Symbol character set. This value specifies that the

Attributes	Description
	characters in the Unicode private use area (U+FF00 to U+FFFF) of the font should be used to display the corresponding characters in the range U+0000 to U+00FF.
	`0x4D` — Specifies a Macintosh (Standard Roman) character set. (IANA name `macintosh`)
	`0x80` — Specifies the JIS character set. (IANA name `shift_jis`)
	`0x81` — Specifies the Hangul character set. (IANA name `ks_c-5601-1987`)
	`0x82` — Specifies a Johab character set. (IANA name `KS_C-5601-1992`)
	`0x86` — Specifies the GB-2312 character set. (IANA name GBK)
	`0x88` — Specifies the Chinese Big Five character set. (IANA name `Big5`)
	`0xA1` — Specifies a Greek character set. (IANA name `windows-1253`)
	`0xA2` — Specifies a Turkish character set. (IANA name `iso-8859-9`)
	`0xA3` — Specifies a Vietnamese character set. (IANA name `windows-1258`)
	`0xB1` — Specifies a Hebrew character set. (IANA name `windows-1255`)
	`0xB2` — Specifies an Arabic character set. (IANA name `windows-1256`)
	`0xBA` — Specifies a Baltic character set. (IANA name `windows-1257`)
	`0xCC` — Specifies a Russian character set. (IANA name `windows-1251`)
	`0xDE` — Specifies a Thai character set. (IANA name `windows-874`)
	`0xEE` — Specifies an Eastern European character set. (IANA name `windows-1250`)
	`0xFF` — Specifies an OEM character set not defined by ISO/IEC 29500.
	Any other value — Application-defined, can be ignored.
	[*Example*: Consider the following value for an attribute of type ST_UCharHexNumber: `<… w:val="BE"/>` This value is permitted, as it contains two hexadecimal digits, an encoding of an octet of the actual decimal number value. *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (Part 1, §17.18.98).

14.5 Numbering

14.5.1 pict (Picture Numbering Symbol Properties)

This element specifies the properties for a picture which shall be used as a picture numbering symbol in a given document, using the VML syntax.

[*Example*: Consider the WordprocessingML below illustrating the usage of the pict element in a document containing a single picture numbering symbol:

```
<w:numPicBullet w:numPicBulletId="0">
  <w:pict>
    <v:shapetype id="_x0000_t75" coordsize="21600,21600" o:spt="75"
o:preferrelative="t" path="m@4@5l@4@11@9@11@9@5xe" filled="f" stroked="f">
      <v:stroke joinstyle="miter" />
      <v:formulas>
        <v:f eqn="if lineDrawn pixelLineWidth 0" />
        <v:f eqn="sum @0 1 0" />
        <v:f eqn="sum 0 0 @1" />
        <v:f eqn="prod @2 1 2" />
        <v:f eqn="prod @3 21600 pixelWidth" />
        <v:f eqn="prod @3 21600 pixelHeight" />
        <v:f eqn="sum @0 0 1" />
        <v:f eqn="prod @6 1 2" />
        <v:f eqn="prod @7 21600 pixelWidth" />
        <v:f eqn="sum @8 21600 0" />
        <v:f eqn="prod @7 21600 pixelHeight" />
        <v:f eqn="sum @10 21600 0" />
      </v:formulas>
      <v:path o:extrusionok="f" gradientshapeok="t" o:connecttype="rect" />
      <o:lock v:ext="edit" aspectratio="t" />
    </v:shapetype>
    <v:shape id="_x0000_i1029" type="#_x0000_t75"
style="width:11.25pt;height:11.25pt" o:bullet="t">
      <v:imagedata r:id="rId1" o:title="sample picture" />
    </v:shape>
  </w:pict>
</w:numPicBullet>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Picture) is located in §A.1. *end note*]

14.5.2 legacy (Legacy Numbering Level Properties)

This element specifies that a given numbering level is from an earlier word processing application which did not support the full richness of the numbering properties supported by WordprocessingML.

These properties shall be used to render any numbered paragraph which references this numbering level if the legacy attribute is set. [*Note*: Using this element in generated WordprocessingML documents is not recommended, as updated numbering structures in WordprocessingML should be used in its place. This element is provided solely to save and roundtrip the numbering properties of legacy word processing products in WordprocessingML such that they are recreated if the document is resaved in an older word processor format. *end note*]

[*Example*: Consider the following WordprocessingML numbering level:

```
<w:lvl w:ilvl="0">
  …
  <w:legacy w:legacySpace="820" w:legacyIndent="960" />
  <w:lvlJc w:val="start" />
  <w:pPr>
    <w:ind w:start="360" w:hanging="360" />
  </w:pPr>
</w:lvl>
```

This level has the legacy element present, therefore the legacy numbering level properties must be used to format all paragraphs which reference this level. *end example*]

Attributes	Description
legacy (Use Legacy Numbering Properties)	Specifies whether the legacy numbering properties present for this numbering level shall be used to format the numbering for any paragraph which references it. A value of `on`, `1`, or `true` for this attribute value specifies that the legacy numbering properties shall be applied. This is the default value for this attribute, and is implied when the attribute is omitted. A value of `off`, `0`, or `false` for this attribute value specifies that the legacy numbering properties shall not be used, and shall be explicitly turned off. [*Example*: For example, consider the set of legacy numbering properties from a document: `<w:legacy w:legacy="off" w:legacySpace="820" w:legacyIndent="960" />` This set of legacy properties are explicitly not used when processing the numbering level via the fact that the legacy attribute is turned `off` for this example. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (Part 1, §22.9.2.7).

Attributes	Description
legacyIndent (Legacy Indent)	Specifies the indentation which shall be applied to a legacy numbering symbol from the text margin of the document. This value is specified in twentieths of a point. If this attribute is not present, then no indentation shall be applied with respect to the margin. [*Example*: For example, consider the set of legacy numbering properties from a document: `<w:legacy w:legacySpace="820" w:legacyIndent="960" />` This set of legacy properties specify that there must be exactly 960 twentieths of a point (⅔ of an inch) between the text margin and the start of the numbering on the paragraph. *end example*] The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (Part 1, §17.18.81).
legacySpace (Legacy Spacing)	Specifies the indentation which shall be applied between a legacy numbering symbol and the accompanying text of the associated paragraph in the document. This value is specified in twentieths of a point. If this attribute is not present, then no indentation shall be applied with respect to the paragraph text. [*Example*: For example, consider the set of legacy numbering properties from a document: `<w:legacy w:legacySpace="820" w:legacyIndent="960" />` This set of legacy properties specify that there must be exactly 860 twentieths of a point between the end of the numbering on the paragraph and the associated paragraph text. *end example*] The possible values for this attribute are defined by the ST_TwipsMeasure simple type (Part 1, §22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LvlLegacy) is located in §A.1. *end note*]

14.6 Annotations

14.6.1 Revisions

14.6.1.1 numberingChange (Previous Numbering Field Properties)

This element specifies the previous state of the numbering displayed by a LISTNUM field (Part 1, §17.16.5.33) within a WordprocessingML document when additional LISTNUM fields are added and revisions are being tracked.

[*Rationale*: The legacy numbering mechanism provided by the LISTNUM field relies on the presence of fields in the run content of the document, rather than being a paragraph property (as numbering typically is represented). For this reason, these fields must store their previous state as a unique revision type on the field character of the numbering field. *end rationale*]

If this element is supplied for a field which is not of type LISTNUM as defined by its field codes (Part 1, §17.16.5), then this property shall be ignored.

[*Example*: Consider the following paragraph containing a single LISTNUM field, as follows:

Some 1. text

If another LISTNUM field is added before it in the document, resulting in its evaluation to a different number, as follows:

Some ~~1.~~2. text

This revision to the field result would be stored as follows in the WordprocessingML:

```
<w:fldChar w:fldCharType="begin">
  <w:numberingChange w:id="0" … w:original="1." />
</w:fldChar>
<w:r>
  <w:instrText>LISTNUM</w:instrText>
</w:r>
<w:fldChar w:fldCharType="separate"/>
<w:r>
  <w:t>2.</w:t>
</w:r>
<w:fldChar w:fldCharType="end" />
```

The numberingChange element specifies that the numbering resulting from this LISTNUM field was modified and this change was tracked as a revision. The previous numbering result of 1. is cached in the original attribute. *end example*]

For numbering fields, the original attribute shall specify the previous numbering displayed by the parent LISTNUM field within a WordprocessingML document. This information is a performance-enhancing cache of the state of the numbering before the revision to allow applications to show the previous state without having to recalculate all of the LISTNUM fields in the document.

If this attribute is omitted, then no previous numbering value is implied and applications can choose to calculate this value, or display no previous numbering value.

[*Example*: Consider the following paragraph containing a single LISTNUM field with a revision, as follows:

Some ~~1.~~2. text

This revision to the field result would be stored as follows in the WordprocessingML:

```
<w:fldChar w:fldCharType="begin">
  <w:numberingChange w:id="0" … w:original="1." />
</w:fldChar>
```

The original attribute specifies that the previous numbering value of the field was `1.` *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (Part 1, §22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500.
	
	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_DateTime simple type (Part 1, §17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element. If this attribute is omitted, then the document is non-conformant.

Attributes	Description

	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`  …`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (Part 1, §17.18.10).

original (Previous Numbering Value)	Specifies the previous numbering displayed by the parent numbering change revision. Its format is specified by the parent element.
	
	If this attribute is omitted, then no previous numbering value is implied and applications can choose to calculate this value, or display no previous numbering value.
	
	[*Example*: Consider the following paragraph containing a single LISTNUM field with a revision, as follows:
	
	Some ~~1.~~2. text
	
	This revision to the field result would be stored as follows in the WordprocessingML:
	
	`<w:fldChar w:fldCharType="begin">`
	`  <w:numberingChange w:id="0" … w:original="1." />`
	`</w:fldChar>`
	
	The original attribute specifies that the previous numbering value of the field was 1. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (Part 1, §22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrackChangeNumbering) is located in §A.1. *end note*]

14.6.1.2 numberingChange (Previous Paragraph Numbering Properties)

This element specifies the previous state of the numbering on a paragraph when revisions are being tracked.

[*Rationale*: This mechanism is simply used to provide storage for revisions to numbering produced by legacy word processing applications, and applications are encouraged to use the pPrChange element to store these changes as changes to the paragraph properties instead. *end rationale*]

[*Example*: Consider the following list using Arabic numerals as the numbering, as follows:

1. one
2. two
3. three

Consider a revision where the numbering definition is changed from Arabic numerals to Roman numerals, as follows:

~~1.~~ i. one
~~2.~~ ii. two
~~3.~~ iii. three

This revision to the numbering definition would be stored as follows in the WordprocessingML:

```
<w:p>
    <w:pPr>
          <w:numPr>
                <w:ilvl w:val="0" />
                <w:numId w:val="1" />
                <w:numberingChange w:id="0" … w:original="%1:1:0:." />
          </w:numPr>
    </w:pPr>
    <w:r>
          <w:t>one</w:t>
    </w:r>
</w:p>
<w:p>
    <w:pPr>
          <w:numPr>
                <w:ilvl w:val="0" />
                <w:numId w:val="1" />
                <w:numberingChange w:id="1" … w:original="%1:2:0:." />
          </w:numPr>
    </w:pPr>
    <w:r>
          <w:t>two</w:t>
    </w:r>
</w:p>
<w:p>
    <w:pPr>
          <w:numPr>
                <w:ilvl w:val="0" />
                <w:numId w:val="1" />
```



```
            <w:numberingChange w:id="2" … w:original="%1:3:0:." />
        </w:numPr>
    </w:pPr>
    <w:r>
        <w:t>three</w:t>
    </w:r>
</w:p>
```

The numberingChange element specifies that the numbering definition was modified and this change was tracked as a revision. The previous Arabic numeral numbering definition is cached in the original attribute. *end example*]

For paragraph numbering, the original attribute shall specify the previous numbering definition for an individual paragraph of text within a WordprocessingML document while revisions are being tracked.

The value of original is represented as separate numbering level definitions defined as follows:

```
<%[numbering level]:[nfc value]:[numbering format]:[separator]>[repeat if more
than one level]
```

where

- `numbering level` – The level for which the numbering definition is defined
- `nfc value` – The value of the numbering style at the specific numbering level
- `numbering format` – The nfc value of the numbering format, as referenced in the table below.
- `separator` – The separator used to separate the numbering level definitions

The numbering format values are mapped as follows:

nfc Value	ST_NumberFormat enumeration equivalent
0	decimal
1	upperRoman
2	lowerRoman
3	upperLetter
4	lowerLetter
5	ordinal
6	cardinalText
7	ordinalText
8	hex
9	chicago
10	ideographDigital
11	japaneseCounting
12	Aiueo

nfc Value	ST_NumberFormat enumeration equivalent
13	Iroha
14	decimalFullWidth
15	decimalHalfWidth
16	japaneseLegal
17	japaneseDigitalTenThousand
18	decimalEnclosedCircle
19	decimalFullWidth2
20	aiueoFullWidth
21	irohaFullWidth
22	decimalZero
23	bullet
24	ganada
25	chosung
26	decimalEnclosedFullstop
27	decimalEnclosedParen
28	decimalEnclosedCircleChinese
29	ideographEnclosedCircle
30	ideographTraditional
31	ideographZodiac
32	ideographZodiacTraditional
33	taiwaneseCounting
34	ideographLegalTraditional
35	taiwaneseCountingThousand
36	taiwaneseDigital
37	chineseCounting
38	chineseLegalSimplified
39	chineseCountingThousand
40	Application-defined. Can be ignored.
41	koreanDigital
42	koreanCounting
43	koreanLegal
44	koreanDigital2
45	hebrew1
46	arabicAlpha
47	hebrew2
48	arabicAbjad
49	hindiVowels
50	hindiConsonants

nfc Value	ST_NumberFormat enumeration equivalent
51	hindiNumbers
52	hindiCounting
53	thaiLetters
54	thaiNumbers
55	thaiCounting
56	vietnameseCounting
57	numberInDash
58	russianLower
59	russianUpper
60 or above	Application-defined. Can be ignored.

[*Example*: Consider the following numbered paragraph where the numbering definition has changed while revisions are being tracked, as follows:

~~1.i.1.~~1. Three

This revision to the numbered paragraph would be stored as follows in the WordprocessingML:

```
<w:numPr>
  …
  <w:numberingChange … w:original="%1:1:0:.%2:1:2:.%3:1:0:." />
</w:numPr>
```

In the above example there are three levels in the original numbering definition, thus three numbering level definitions are needed to represent the original numbering definition.

The first level is specified by `%1`, and says that it was number value `1` in the nfc format `0` (`arabic`).

The original attribute specifies that the previous numbering definition was made up of three levels whose value was `1.i.1.`. *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`

Attributes	Description
	`</…>` The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*] The possible values for this attribute are defined by the ST_String simple type (Part 1, §22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500.
	
	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_DateTime simple type (Part 1, §17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (Part 1, §17.18.10).

original (Previous Numbering Value)	Specifies the previous numbering displayed by the parent numbering change revision. Its format is specified by the parent element.

Attributes	Description

	If this attribute is omitted, then no previous numbering value is implied and applications can choose to calculate this value, or display no previous numbering value.
	
	[*Example*: Consider the following paragraph containing a single `LISTNUM` field with a revision, as follows:
	
	Some ~~1.~~2. text
	
	This revision to the field result would be stored as follows in the WordprocessingML:
	
	`<w:fldChar w:fldCharType="begin">`
	`  <w:numberingChange w:id="0" … w:original="1." />`
	`</w:fldChar>`
	
	The original attribute specifies that the previous numbering value of the field was `1.` *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (Part 1, §22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrackChangeNumbering) is located in §A.1. *end note*]

14.7 Settings

14.7.1 Legacy Password Hash Algorithm

When a password hash value is stored using the transitional hashing mechanism described in the following subclause, that process shall be done in two stages:

The following steps assume that all words are unsigned, the word size is two bytes, and that bit-level SHL/SHR operations shift in the direction of the highest-order and lowest-order bit, respectively. [*Example*: `0x61 SHR 1` is `0xC2`, as `01100001` shifted one position in the direction of its highest-order bit is `11000010`. *end example*]

The UTF-16LE encoded password shall be hashed using the following algorithm (if there is a leading BOM character (U+FEFF) in the encoded password it is removed before hash calculation):

- Passwords of 15 or fewer characters shall be used in the hash without further change; passwords longer than 15 characters shall be truncated to 15 characters.
- Construct a new NULL-terminated string consisting of single-byte values using the algorithm described by the following bullet. The input to this step should be the series of UTF-16 characters defined above:
    - Get the single-byte values by iterating through the Unicode characters of the truncated password. For each character, if the low byte is not equal to 0, take it. Otherwise, take the high byte.

- From now on, the single-byte character string is used.
- If the password is empty, return 0.
- Compute the high-order word of the new key:
  - Initialize from the initial code array (see below), depending on the password's length. For each character in the password:
    - For every bit in the character, starting with the least significant and progressing to (but excluding) the most significant, if the bit is set, XOR the key's high-order word with the corresponding word from the encryption matrix
- Compute the low-order word of the new key:
  - Initialize with 0
    - For each character in the password, going backwards, low-order word = (((low-order word SHR 14) AND 0x0001) OR (low-order word SHL 1) AND 0x7FFF)) XOR character
    - Lastly, low-order word = (((low-order word SHR 14) AND 0x0001) OR (low-order word SHL 1) AND 0x7FFF)) XOR password length XOR 0xCE4B.

Initial code array

The initial code array contains the initial values for the key's high-order word. The initial value depends on the length of the password, as follows:

Password length	Initial value for the key's high-order word
1	0xE1F0
2	0x1D0F
3	0xCC9C
4	0x84C0
5	0x110C
6	0x0E10
7	0xF1CE
8	0x313E
9	0x1872
10	0xE139
11	0xD40F
12	0x84F9
13	0x280C
14	0xA96A
15	0x4EC3

Encryption matrix

The encryption matrix contains codes used during the calculation of the key's high-order word. As described in the algorithm above, for every bit of the password's characters, if the bit is set, a corresponding value is taken from this encryption matrix and is used to XOR the key's high-order word with it. Each row in the encryption matrix corresponds to a single character from the password, and each of the seven columns corresponds to a particular bit (0-6) in this character.

The values are taken in such a way so that the last character of the password uses the last row in the encryption matrix. The next-to-last character uses the next-to-last row in the matrix, and so on. This means that the beginning of the matrix might be unused, depending on the length of the password.

	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6
Last-14	0xAEFC	0x4DD9	0x9BB2	0x2745	0x4E8A	0x9D14	0x2A09
Last-13	0x7B61	0xF6C2	0xFDA5	0xEB6B	0xC6F7	0x9DCF	0x2BBF
Last-12	0x4563	0x8AC6	0x05AD	0x0B5A	0x16B4	0x2D68	0x5AD0
Last-11	0x0375	0x06EA	0x0DD4	0x1BA8	0x3750	0x6EA0	0xDD40
Last-10	0xD849	0xA0B3	0x5147	0xA28E	0x553D	0xAA7A	0x44D5
Last-9	0x6F45	0xDE8A	0xAD35	0x4A4B	0x9496	0x390D	0x721A
Last-8	0xEB23	0xC667	0x9CEF	0x29FF	0x53FE	0xA7FC	0x5FD9
Last-7	0x47D3	0x8FA6	0x0F6D	0x1EDA	0x3DB4	0x7B68	0xF6D0
Last-6	0xB861	0x60E3	0xC1C6	0x93AD	0x377B	0x6EF6	0xDDEC
Last-5	0x45A0	0x8B40	0x06A1	0x0D42	0x1A84	0x3508	0x6A10
Last-4	0xAA51	0x4483	0x8906	0x022D	0x045A	0x08B4	0x1168
Last-3	0x76B4	0xED68	0xCAF1	0x85C3	0x1BA7	0x374E	0x6E9C
Last-2	0x3730	0x6E60	0xDCC0	0xA9A1	0x4363	0x86C6	0x1DAD
Last-1	0x3331	0x6662	0xCCC4	0x89A9	0x0373	0x06E6	0x0DCC
Last	0x1021	0x2042	0x4084	0x8108	0x1231	0x2462	0x48C4

[*Example*: Consider a password which has been supplied - the string "Example". It is already under 15 characters, so truncation does not affect it. It is then converted to a string of single-byte characters.

- The password is 7 characters long, so, from the initial code array, the initial value for the key's high-order word is 0xF1CE.
- The key's high-order word is then computed further depending on the password's characters:
    - The first character is 'E' (0x45). This is the first character of a 7-character password, so its corresponding row in the encryption matrix is "Last-6".

    - Bit 0 is set, therefore the key's high-order word is combined (via XOR) with the corresponding value for Bit 0 on row "Last-6", which is 0xB861. The new result is 0xF1CE XOR 0xB861 = 0x49AF.
    - Bit 2 is set, so the key's high-order word is XOR-ed with the corresponding value for Bit 2 on row "Last-6", which is 0xC1C6. The new result is 0x49AF XOR 0xC1C6 = 0x8869.
    - This process is repeated for each bit.
  - The next character is 'x' (0x78). Its corresponding row in the encryption matrix is "Last-5".
    - Bit 3 is set. The value for Bit 3 on row "Last-5" in the encryption matrix is 0x0D42. The current value for the key's high-order byte is 0x5585, so the new one should be 0x5585 XOR 0x0D42 = 0x58C7.
    - This process is repeated for each bit.
  - This process is repeated for all characters.
- After the last character has been processed, the above step produced 0x64CE for the key's high-order word. Now the low-order word needs to be calculated:
  - The initial value is 0.
  - It is then calculated using the password:
    - The last character of the password is 'e' (0x65), so, by the formula, low-order word = (((low-order word SHR 14) AND 0x0001) OR ((low-order word SHL 1) AND 0x7FFF)) XOR 'e' = (((0 SHR 14) AND 0x0001) OR ((0 SHL 1) AND 0x7FFF)) XOR 0x65 = 0x0065.
    - The next to last character of the password is 'l' (0x6C). Again, by the formula, (((0x0065 SHR 14) AND 0x0001) OR ((0x0065 SHL 1) AND 0x7FFF)) XOR 0x6C = (0x0000 OR 0x00CA) XOR 0x6C = 0x00CA XOR 0x6C = 0x00A6.
    - This process is repeated for each character.
  - After the password's first character has been processed, we have 0x1199 for the key's low-order word. Lastly, the password's length is combined into it: low-order word = (((0x1199 SHR 14) AND 0x0001) OR ((0x1199 SHL 1) AND 0x7FFF)) XOR 0x0007 XOR 0xCE4B = 0x2332 XOR 0x0007 XOR 0xCE4B = 0x2335 XOR 0xCE4B = 0xED7E.
- The end result for the key is 0x64CEED7E.

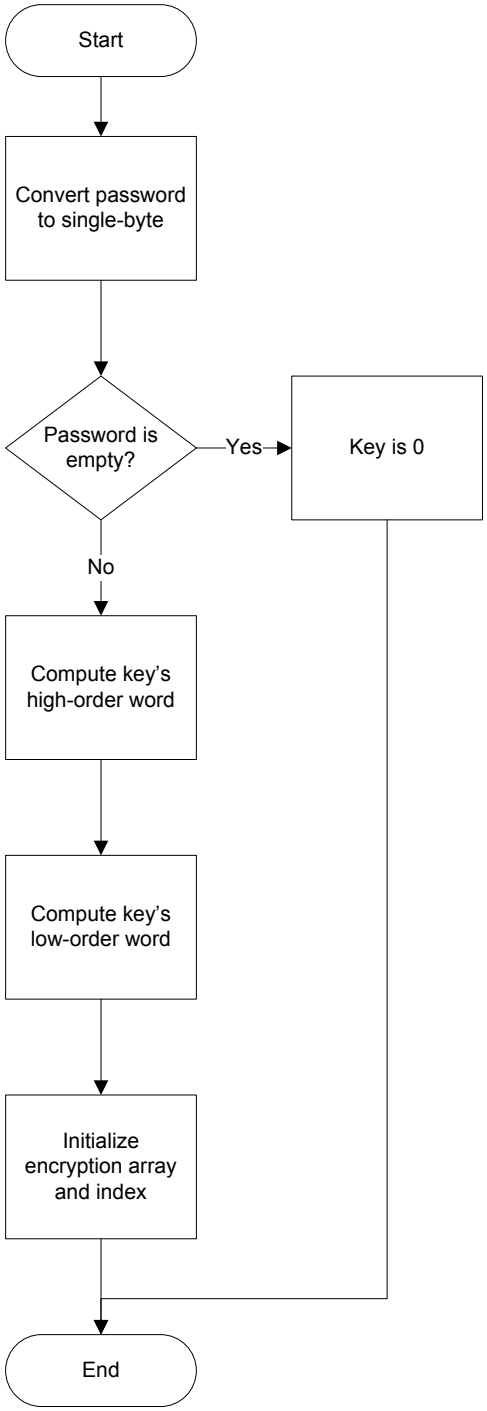
end example]

[*Rationale*: This pre-processing step is necessary for compatibility with legacy word processing applications which hashed their password solely using this mechanism. *end rationale*]

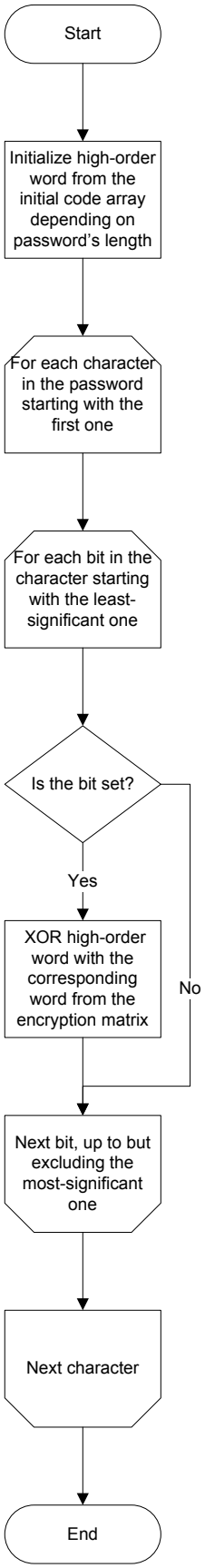
Second, the byte order of the result shall be reversed [*Example*: 0x64CEED7E becomes 7EEDCE64. *end example*], and that value shall be hashed as defined by the attribute values.

[*Note*: The algorithm above can be stated as follows using diagrams:

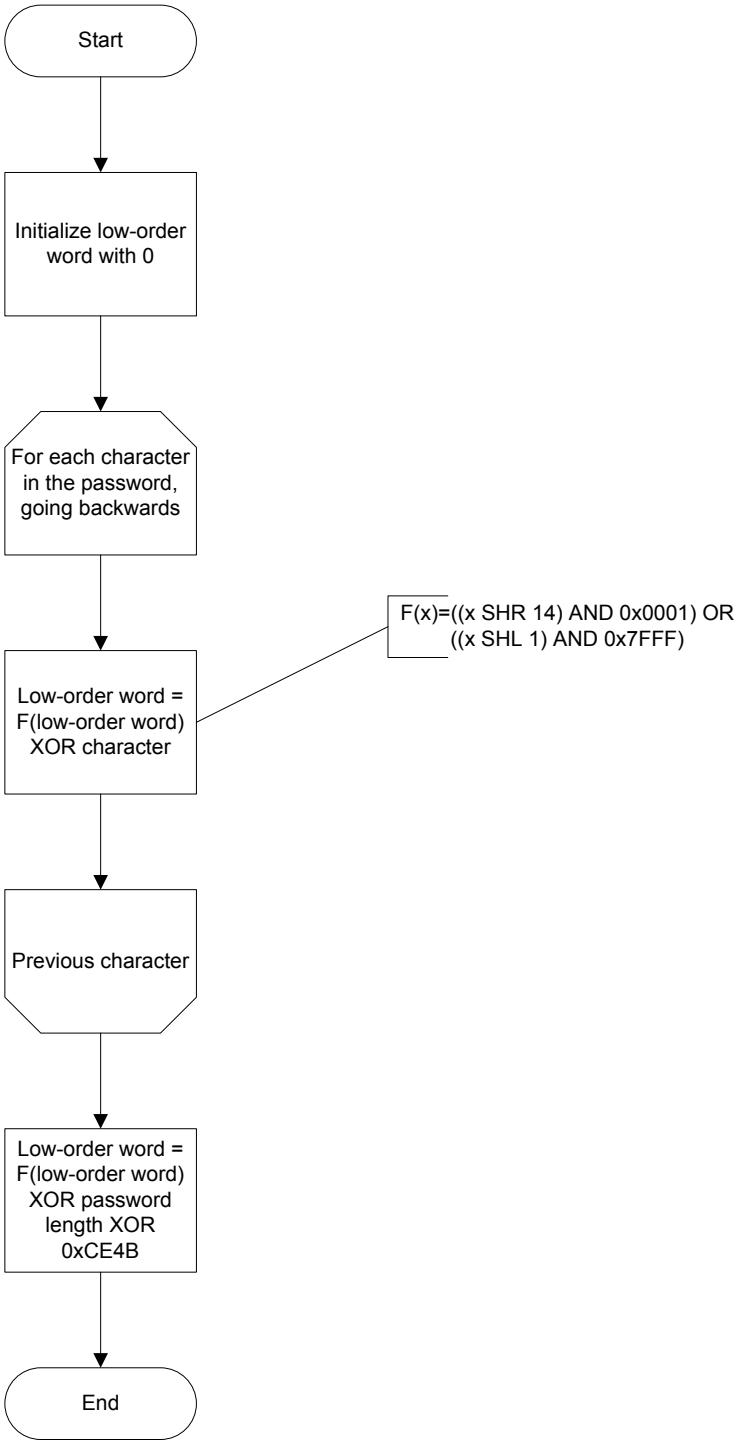
Calculate Key



Compute Key's High-Order Word



Compute Key's Low-Order Word



end note]

[*Example*: Consider a WordprocessingML document which specifies that applications must not allow any modifications to this document other than the addition of comments. This requirement would be specified using the following WordprocessingML in the document settings:

```
<w:documentProtection w:edit="comments" w:enforcement="true" …
  w:cryptAlgorithmClass="hash" w:cryptAlgorithmType="typeAny"
  w:cryptAlgorithmSid="1" w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />
```

The documentProtection element has an edit attribute value of `comments`, specifying that the only modification allowed should be comments, the enforcement attribute has a value of `true`, specifying that the document protection specified is to be enforced on the given document. Finally, in order for the hosting application to stop enforcement of the document protection applied to the document, the hosting application would have to be provided with a password that the hosting application would then hash, compare to the value of the hash attribute (`9oN7nWkCAyEZib1RomSJTjmPpCY=`), and if the two values matched, halt enforcement of any document protection. *end example*]

14.7.2 Document Settings

14.7.2.1 hdrShapeDefaults (Default Properties for VML Objects in Header and Footer)

This element specifies the default parameters for object using the VML syntax (§19.1) inserted in the header and footer of a WordprocessingML document. The definition and semantics of these parameters is described in the VML - Office Drawing subclause (§19.2) of ISO/IEC 29500.

If this element is omitted, then no default properties are applied to VML objects in the header and footer of this document.

[*Example*: Consider a WordprocessingML document whose document settings contain the following markup:

```
<w:hdrShapeDefaults>
  <o:shapedefaults v:ext="edit" spidmax="2050" fillcolor="none [3207]"
strokecolor="none [3041]">
    <v:fill color="none [3207]" />
    <v:stroke color="none [3041]" weight="3pt" />
    <v:shadow on="t" type="perspective" color="none [1607]" opacity=".5"
offset="1pt" offset2="1pt" />
  </o:shapedefaults>
  <o:shapelayout v:ext="edit">
    <o:idmap v:ext="edit" data="2" />
  </o:shapelayout>
</w:hdrShapeDefaults>
```

The hdrShapeDefaults element specifies a set of shape defaults which must be applied to the set of all shapes present in the header and footer of this document. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeDefaults) is located in §A.1. *end note*]

14.7.2.2 shapeDefaults (Default Properties for VML Objects in Main Document)

This element specifies the default parameters for object using the VML syntax (§19.1) inserted in the body (the main document story, comments, footnotes, and endnotes) of the WordprocessingML document. The definition and semantics of these parameters is described in the VML - Office Drawing subclause (§19.2) of ISO/IEC 29500.

If this element is omitted, then no default properties are applied to VML objects in the body of this document.

[*Example*: Consider a WordprocessingML document whose document settings contain the following markup:

```
<w:shapeDefaults>
  <o:shapedefaults v:ext="edit" spidmax="1026" />
  <o:shapelayout v:ext="edit">
    <o:idmap v:ext="edit" data="1" />
  </o:shapelayout>
</w:shapeDefaults>
```

The shapeDefaults element specifies a set of shape defaults which must be applied to the set of all shapes present in the body document. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeDefaults) is located in §A.1. *end note*]

14.7.2.3 Additional attributes for documentProtection element (Part 1, §17.15.1.29)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description
algIdExt (Cryptographic Algorithm Extensibility)	Specifies that a cryptographic algorithm which was not defined by ISO/IEC 29500 has been used to generate the hash value stored with this document. This value, when present, shall be interpreted based on the value of the algIdExtSource attribute in order to determine the algorithm used, which shall be application-defined. [*Rationale*: This extensibility affords the fact that with exponentially increasing computing power, documents created in the future might need to utilize as yet undefined hashing algorithms in order to remain secure. *end rationale*] If this value is present, the cryptAlgorithmClass, cryptAlgorithmType, and cryptAlgorithmSid attribute values shall be ignored in favor of the algorithm defined by this attribute. [*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements: `<… w:algIdExt="0000000A"`

Attributes	Description

	`w:algIdExtSource="futureCryptography"`
	`w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The algIdExt attribute value of `0000000A` specifies that the algorithm with hex code `A` must be used as defined by the `futureCryptography` application. *end example*]
	
	The possible values for this attribute are defined by the ST_LongHexNumber simple type (Part 1, §17.18.50).

algIdExtSource (Algorithm Extensibility Source)	Specifies the application which defined the algorithm value specified by the algIdExt attribute.
	
	[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:
	
	`<… w:algIdExt="0000000A"`
	`w:algIdExtSource="futureCryptography"`
	`w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The algIdExtSource attribute value of `futureCryptography` specifies that the algorithm used here was published by the `futureCryptography` application. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (Part 1, §22.9.2.13).

cryptAlgorithmClass (Cryptographic Algorithm Class)	Specifies the class of cryptographic algorithm used by this protection. [*Note*: The initial version of ISO/IEC 29500 only supports a single version - `hash` - but future versions can expand this as necessary. *end note*]
	
	[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:
	
	`<… w:cryptAlgorithmClass="hash"`
	`w:cryptAlgorithmType="typeAny"`
	`w:cryptAlgorithmSid="1"`
	`w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The cryptAlgorithmClass attribute value of `hash` specifies that the algorithm used for the password is a hashing algorithm. *end example*]
	
	The possible values for this attribute are defined by the ST_AlgClass simple type (§20.1.2.1).

cryptAlgorithmSid (Cryptographic Hashing Algorithm)	Specifies the specific cryptographic hashing algorithm which shall be used along with the salt attribute and user-supplied password in order to compute a hash value for comparison. The possible values for this attribute shall be interpreted as follows:

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Algorithm</th></tr>
<tr><td>1</td><td>MD2</td></tr>
<tr><td>2</td><td>MD4</td></tr>
<tr><td>3</td><td>MD5</td></tr>
<tr><td>4</td><td>SHA-1</td></tr>
<tr><td>5</td><td>MAC</td></tr>
<tr><td>6</td><td>RIPEMD</td></tr>
<tr><td>7</td><td>RIPEMD-160</td></tr>
<tr><td>8</td><td>Undefined. Shall not be used.</td></tr>
<tr><td>9</td><td>HMAC</td></tr>
<tr><td>10</td><td>Undefined. Shall not be used.</td></tr>
<tr><td>11</td><td>Undefined. Shall not be used.</td></tr>
<tr><td>12</td><td>SHA-256</td></tr>
<tr><td>13</td><td>SHA-384</td></tr>
<tr><td>14</td><td>SHA-512</td></tr>
<tr><td>Any other value</td><td>Undefined. Shall not be used.</td></tr>
</table>

[Example: Consider a WordprocessingML document with the following information stored in one of its protection elements:

<pre><… w:cryptAlgorithmClass="hash"
  w:cryptAlgorithmType="typeAny"
  w:cryptAlgorithmSid="4"
  w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" /></pre>

The cryptAlgorithmSid attribute value of 4 specifies that the SHA-1 hashing algorithm must be used to generate a hash from the user-defined password. end example]

The possible values for this attribute are defined by the ST_DecimalNumber simple type (Part 1, §17.18.10).
</td></tr>
<tr><td>cryptAlgorithmType (Cryptographic Algorithm Type)</td><td>
Specifies the type of cryptographic algorithm used by this protection. [Note: The initial version of ISO/IEC 29500 only supports a single algorithm type - typeAny - but future versions can expand this as necessary. end note]

[Example: Consider a WordprocessingML document with the following information stored in one of its protection elements:

<pre><… w:cryptAlgorithmClass="hash"
  w:cryptAlgorithmType="typeAny"</pre>
</td></tr>
</table>

Attributes	Description

	`w:cryptAlgorithmSid="1"`
	`w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The cryptAlgorithmType attribute value of `typeAny` specifies that any type of algorithm might have been used for the password. *end example*]
	
	The possible values for this attribute are defined by the ST_AlgType simple type (§20.1.2.2).

cryptProvider (Cryptographic Provider)	Specifies the cryptographic provider which was used to generate the hash value stored in this document. If the user provided a cryptographic provider which was not the system's built-in provider, then that provider shall be stored here so it can subsequently be used if available.
	
	If this attribute is omitted, then the built-in cryptographic provider on the system shall be used.
	
	[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:
	
	`<… w:cryptProvider="Krista'sProvider"`
	`w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The cryptProvider attribute value of `Krista'sProvider` specifies that the cryptographic provider with name "Krista's Provider" must be used if available. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (Part 1, §22.9.2.13).

cryptProviderType (Cryptographic Provider Type)	Specifies the type of cryptographic provider to be used.
	
	[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:
	
	`<… w:cryptProviderType="rsaAES"`
	`w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The cryptProviderType attribute value of `rsaAES` specifies that the cryptographic provider type must be an Advanced Encryption Standard provider. *end example*]
	
	The possible values for this attribute are defined by the ST_CryptProv simple type (§20.1.2.4).

cryptProviderTypeExt (Cryptographic Provider Type Extensibility)	Specifies that a cryptographic provider type which was not defined by ISO/IEC 29500 has been used to generate the hash value stored with this document. This value, when present, shall be interpreted based on the value of the cryptProviderTypeExtSource attribute in order to determine the provider type used,

Attributes	Description

	which shall be application-defined. [*Rationale*: This extensibility affords the fact that with exponentially increasing computing power, documents created in the future might need to utilize as yet undefined cryptographic provider types in order to remain secure. *end rationale*]
	
	If this value is present, the cryptProviderType attribute value shall be ignored in favor of the provider type defined by this attribute.
	
	[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:
	
	`<… w:cryptProviderTypeExt="00A5691D"`
	`w:cryptProvideTypeExtSource="futureCryptography"`
	`w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The cryptProviderTypeExt attribute value of `00A5691D` specifies that the provider type associated with hex code `A5691D` must be used as defined by the `futureCryptography` application. *end example*]
	
	The possible values for this attribute are defined by the ST_LongHexNumber simple type (Part 1, §17.18.50).

cryptProviderTypeExtSource (Provider Type Extensibility Source)	Specifies the application which defined the provider type value specified by the cryptProviderTypeExt attribute.
	
	[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:
	
	`<… w:cryptProviderTypeExt="00A5691D"`
	`w:cryptProvideTypeExtSource="futureCryptography"`
	`w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The cryptProvideTypeExtSource attribute value of `futureCryptography` specifies that the provider type used here was published by the `futureCryptography` application. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (Part 1, §22.9.2.13).

cryptSpinCount (Iterations to Run Hashing Algorithm)	Specifies the number of times the hashing function shall be iteratively run (runs using each iteration's result plus a 4 byte value (0-based, little endian) containing the number of the iteration as the input for the next iteration) when attempting to compare a user-supplied password with the value stored in the hash attribute. [*Rationale*: Running the algorithm many times increases the cost of exhaustive search attacks correspondingly. Storing this value allows for the number of iterations to be increased over time to accommodate faster hardware (and hence the ability to run more iterations in less time). *end rationale*]

Attributes	Description

	[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:
	
	`<… w:cryptSpinCount="100000"`
	`  w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The cryptSpinCount attribute value of `100000` specifies that the hashing function must be run one hundred thousand times to generate a hash value for comparison with the hash attribute. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (Part 1, §17.18.10).

hash (Password Hash)	Specifies the hash value for the password stored with this document. This value shall be compared with the resulting hash value after hashing the user-supplied password using the algorithm specified by the preceding attributes and parent XML element, and if the two values match, the protection shall no longer be enforced.
	
	If this value is omitted, then no password shall be associated with the protection, and it can be turned off without supplying any password.
	
	[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:
	
	`<… w:cryptAlgorithmClass="hash"`
	`  w:cryptAlgorithmType="typeAny"`
	`  w:cryptAlgorithmSid="1"`
	`  w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The hash attribute value of `9oN7nWkCAyEZib1RomSJTjmPpCY=` specifies that the user-supplied password must be hashed using the pre-processing defined by the parent element (if any) followed by the SHA-1 algorithm (specified via the cryptAlgorithmSid attribute value of 1) and that the resulting hash value must be `9oN7nWkCAyEZib1RomSJTjmPpCY=` for the protection to be disabled. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.

salt (Salt for Password Verifier)	Specifies the salt which was prepended to the user-supplied password before it was hashed using the hashing algorithm defined by the preceding attribute values to generate the hash attribute, and which shall also be prepended to the user-supplied password before attempting to generate a hash value for comparison. A *salt* is a random string which is added to a user-supplied password before it is hashed in order to prevent a malicious party from pre-calculating all possible password/hash combinations and simply using those precalculated values (often referred to as a "dictionary attack"). If this attribute is omitted, then no salt shall be prepended to the user-supplied password before it is hashed for comparison with the stored hash value.

Attributes	Description

	[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:
	
	`<… w:salt="ZUdHa+D8F/OAKP3I7ssUnQ=="`
	`w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The salt attribute value of `ZUdHa+D8F/OAKP3I7ssUnQ==` specifies that the user-supplied password must have this value prepended before it is run through the specified hashing algorithm to generate a resulting hash value for comparison. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.

14.7.2.4 Additional attribute for stylePaneFormatFilter element (Part 1, §17.15.1.85)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description
val (Bitmask of Suggested Filtering Options)	Specifies a bitmask of the following filtering options: **Value** — **Description** `0x0001` — Specifies that all styles present in the styles part should be displayed in the list of document styles. `0x0002` — Specifies that only styles with the customStyle attribute should be displayed in the list of document styles. `0x0004` — Specifies that all latent styles should be displayed in the list of document styles. `0x0008` — Specifies that only styles used in the document should be displayed in the list of document styles. `0x0010` — Undefined. Shall not be used. `0x0020` — Specifies that heading styles (styles with a styleId of `Heading1` to `Heading9`) should be displayed in the list of document styles when the previous style is used in the document and/or is present in the styles part. `0x0040` — Specifies that numbering styles should be displayed in the list of document styles. `0x0080` — Specifies that table styles should be displayed in the list of document styles. `0x0100` — Specifies that all unique forms of run-level direct formatting should be displayed in the list of document styles as though they were each a unique style. `0x0200` — Specifies that all unique forms of paragraph-level direct formatting should

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>be displayed in the list of document styles as though they were each a unique style.</td></tr>
<tr><td>0x0400</td><td>Specifies that all unique forms of direct formatting of numbering data should be displayed in the list of document styles as though they were each a unique style.</td></tr>
<tr><td>0x0800</td><td>Specifies that all unique forms of direct formatting of tables should be displayed in the list of document styles as though they were each a unique style.</td></tr>
<tr><td>0x1000</td><td>Specifies that a style should be present which removes all formatting and styles from text.</td></tr>
<tr><td>0x2000</td><td>Specifies that heading styles with a styleId of Heading1 to Heading3 should always be displayed in the list of document styles.</td></tr>
<tr><td>0x4000</td><td>Specifies that styles should only be shown the semiHidden element (Part 1, §17.7.4.16) is false and the hidden element (Part 1, §17.7.4.4) is false.</td></tr>
<tr><td>0x8000</td><td>Specifies that primary names for styles should not be shown if an alternate name using the name element (Part 1, §17.7.4.9) exists.</td></tr>
<tr><td>Any other value</td><td>Undefined. Shall not be used.</td></tr>
</table>

[Example: Consider a document with the following value in its document settings:

<pre><w:stylePaneFormatFilter w:val="2002" /></pre>

The val attribute specifies two suggested filter options for the list of document styles:
• Only custom styles should be shown (0002)
• Heading styles with a styleId of Heading1 to Heading3 should always be displayed in the list (2000)

end example]

The possible values for this attribute are defined by the ST_ShortHexNumber simple type (Part 1, §17.18.79).
</td></tr>
</table>

14.7.2.5 Additional attributes for writeProtection element (Part 1, §17.15.1.93)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description
algIdExt (Cryptographic Algorithm	Specifies that a cryptographic algorithm which was not defined by ISO/IEC 29500 has been used to generate the hash value stored with this document.

Attributes	Description

Extensibility)	This value, when present, shall be interpreted based on the value of the algIdExtSource attribute in order to determine the algorithm used, which shall be application-defined. [*Rationale*: This extensibility affords the fact that with exponentially increasing computing power, documents created in the future might need to utilize as yet undefined hashing algorithms in order to remain secure. *end rationale*]
	
	If this value is present, the cryptAlgorithmClass, cryptAlgorithmType, and cryptAlgorithmSid attribute values shall be ignored in favor of the algorithm defined by this attribute.
	
	[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:
	
	`<… w:algIdExt="0000000A"`
	`  w:algIdExtSource="futureCryptography"`
	`  w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The algIdExt attribute value of `0000000A` specifies that the algorithm with hex code `A` must be used as defined by the `futureCryptography` application. *end example*]
	
	The possible values for this attribute are defined by the ST_LongHexNumber simple type (Part 1, §17.18.50).

algIdExtSource (Algorithm Extensibility Source)	Specifies the application which defined the algorithm value specified by the algIdExt attribute.
	
	[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:
	
	`<… w:algIdExt="0000000A"`
	`  w:algIdExtSource="futureCryptography"`
	`  w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The algIdExtSource attribute value of `futureCryptography` specifies that the algorithm used here was published by the `futureCryptography` application. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (Part 1, §22.9.2.13).

cryptAlgorithmClass (Cryptographic Algorithm Class)	Specifies the class of cryptographic algorithm used by this protection. [*Note*: The initial version of ISO/IEC 29500 only supports a single version - `hash` - but future versions can expand this as necessary. *end note*]
	
	[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:
	
	`<… w:cryptAlgorithmClass="hash"`
	`  w:cryptAlgorithmType="typeAny"`

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>

```
w:cryptAlgorithmSid="1"
w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />
```

The cryptAlgorithmClass attribute value of hash specifies that the algorithm used for the password is a hashing algorithm. *end example*]

The possible values for this attribute are defined by the ST_AlgClass simple type (§20.1.2.1).
</td></tr>
<tr><td>cryptAlgorithmSid (Cryptographic Hashing Algorithm)</td><td>

Specifies the specific cryptographic hashing algorithm which shall be used along with the salt attribute and user-supplied password in order to compute a hash value for comparison.

The possible values for this attribute shall be interpreted as follows:

<table>
<tr><th>Value</th><th>Algorithm</th></tr>
<tr><td>1</td><td>MD2</td></tr>
<tr><td>2</td><td>MD4</td></tr>
<tr><td>3</td><td>MD5</td></tr>
<tr><td>4</td><td>SHA-1</td></tr>
<tr><td>5</td><td>MAC</td></tr>
<tr><td>6</td><td>RIPEMD</td></tr>
<tr><td>7</td><td>RIPEMD-160</td></tr>
<tr><td>8</td><td>Undefined. Shall not be used.</td></tr>
<tr><td>9</td><td>HMAC</td></tr>
<tr><td>10</td><td>Undefined. Shall not be used.</td></tr>
<tr><td>11</td><td>Undefined. Shall not be used.</td></tr>
<tr><td>12</td><td>SHA-256</td></tr>
<tr><td>13</td><td>SHA-384</td></tr>
<tr><td>14</td><td>SHA-512</td></tr>
<tr><td>Any other value</td><td>Undefined. Shall not be used.</td></tr>
</table>

[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:

```
<… w:cryptAlgorithmClass="hash"
  w:cryptAlgorithmType="typeAny"
  w:cryptAlgorithmSid="4"
  w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />
```

The cryptAlgorithmSid attribute value of 4 specifies that the SHA-1 hashing algorithm
</td></tr>
</table>

Attributes	Description
	must be used to generate a hash from the user-defined password. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (Part 1, §17.18.10).

cryptAlgorithmType (Cryptographic Algorithm Type)	Specifies the type of cryptographic algorithm used by this protection. [*Note*: The initial version of ISO/IEC 29500 only supports a single algorithm type - `typeAny` - but future versions can expand this as necessary. *end note*]
	
	[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:
	
	`<… w:cryptAlgorithmClass="hash"`
	`w:cryptAlgorithmType="typeAny"`
	`w:cryptAlgorithmSid="1"`
	`w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The cryptAlgorithmType attribute value of `typeAny` specifies that any type of algorithm might have been used for the password. *end example*]
	
	The possible values for this attribute are defined by the ST_AlgType simple type (§20.1.2.2).

cryptProvider (Cryptographic Provider)	Specifies the cryptographic provider which was used to generate the hash value stored in this document. If the user provided a cryptographic provider which was not the system's built-in provider, then that provider shall be stored here so it can subsequently be used if available.
	
	If this attribute is omitted, then the built-in cryptographic provider on the system shall be used.
	
	[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:
	
	`<… w:cryptProvider="Krista'sProvider"`
	`w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The cryptProvider attribute value of `Krista'sProvider` specifies that the cryptographic provider with name "Krista's Provider" must be used if available. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (Part 1, §22.9.2.13).

cryptProviderType (Cryptographic Provider Type)	Specifies the type of cryptographic provider to be used. [*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:

Attributes	Description

	`<… w:cryptProviderType="rsaAES"`
	`w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The cryptProviderType attribute value of `rsaAES` specifies that the cryptographic provider type must be an Advanced Encryption Standard provider. *end example*]
	
	The possible values for this attribute are defined by the ST_CryptProv simple type (§20.1.2.4).

cryptProviderTypeExt (Cryptographic Provider Type Extensibility)	Specifies that a cryptographic provider type which was not defined by ISO/IEC 29500 has been used to generate the hash value stored with this document.
	
	This value, when present, shall be interpreted based on the value of the cryptProviderTypeExtSource attribute in order to determine the provider type used, which shall be application-defined. [*Rationale*: This extensibility affords the fact that with exponentially increasing computing power, documents created in the future might need to utilize as yet undefined cryptographic provider types in order to remain secure. *end rationale*]
	
	If this value is present, the cryptProviderType attribute value shall be ignored in favor of the provider type defined by this attribute.
	
	[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:
	
	`<… w:cryptProviderTypeExt="00A5691D"`
	`w:cryptProvideTypeExtSource="futureCryptography"`
	`w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The cryptProviderTypeExt attribute value of `00A5691D` specifies that the provider type associated with hex code `A5691D` must be used as defined by the `futureCryptography` application. *end example*]
	
	The possible values for this attribute are defined by the ST_LongHexNumber simple type (Part 1, §17.18.50).

cryptProviderTypeExtSource (Provider Type Extensibility Source)	Specifies the application which defined the provider type value specified by the cryptProviderTypeExt attribute.
	
	[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:
	
	`<… w:cryptProviderTypeExt="00A5691D"`
	`w:cryptProvideTypeExtSource="futureCryptography"`
	`w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The cryptProvideTypeExtSource attribute value of `futureCryptography` specifies that the provider type used here was published by the `futureCryptography` application. *end*

Attributes	Description
	example] The possible values for this attribute are defined by the ST_String simple type (Part 1, §22.9.2.13).

cryptSpinCount (Iterations to Run Hashing Algorithm)	Specifies the number of times the hashing function shall be iteratively run (runs using each iteration's result plus a 4 byte value (0-based, little endian) containing the number of the iteration as the input for the next iteration) when attempting to compare a user-supplied password with the value stored in the hash attribute. [*Rationale*: Running the algorithm many times increases the cost of exhaustive search attacks correspondingly. Storing this value allows for the number of iterations to be increased over time to accommodate faster hardware (and hence the ability to run more iterations in less time). *end rationale*]
	
	[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:
	
	`<… w:cryptSpinCount="100000"`
	`  w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The cryptSpinCount attribute value of `100000` specifies that the hashing function must be run one hundred thousand times to generate a hash value for comparison with the hash attribute. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (Part 1, §17.18.10).

hash (Password Hash)	Specifies the hash value for the password stored with this document. This value shall be compared with the resulting hash value after hashing the user-supplied password using the algorithm specified by the preceding attributes and parent XML element, and if the two values match, the protection shall no longer be enforced.
	
	If this value is omitted, then no password shall be associated with the protection, and it can be turned off without supplying any password.
	
	[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:
	
	`<… w:cryptAlgorithmClass="hash"`
	`  w:cryptAlgorithmType="typeAny"`
	`  w:cryptAlgorithmSid="1"`
	`  w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The hash attribute value of `9oN7nWkCAyEZib1RomSJTjmPpCY=` specifies that the user-supplied password must be hashed using the pre-processing defined by the parent element (if any) followed by the SHA-1 algorithm (specified via the cryptAlgorithmSid attribute value of 1) and that the resulting hash value must be `9oN7nWkCAyEZib1RomSJTjmPpCY=` for the protection to be disabled. *end example*]

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.

salt (Salt for Password Verifier)	Specifies the salt which was prepended to the user-supplied password before it was hashed using the hashing algorithm defined by the preceding attribute values to generate the hash attribute, and which shall also be prepended to the user-supplied password before attempting to generate a hash value for comparison. A *salt* is a random string which is added to a user-supplied password before it is hashed in order to prevent a malicious party from pre-calculating all possible password/hash combinations and simply using those precalculated values (often referred to as a "dictionary attack").
	
	If this attribute is omitted, then no salt shall be prepended to the user-supplied password before it is hashed for comparison with the stored hash value.
	
	[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:
	
	`<… w:salt="ZUdHa+D8F/OAKP3I7ssUnQ=="`
	`w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The salt attribute value of `ZUdHa+D8F/OAKP3I7ssUnQ==` specifies that the user-supplied password must have this value prepended before it is run through the specified hashing algorithm to generate a resulting hash value for comparison. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.

14.7.3 Compatibility Settings

The last group of settings stored in WordprocessingML is compatibility settings. *Compatibility Settings* are optional settings used to preserve visual fidelity of documents created in earlier word processing applications. Some of these settings provide affordance for specific behaviors, described in detail below; and others simply instruct applications to mimic the behavior of an existing word processing application.

If compatibility settings are needed, they are stored in the Document Settings part.

It is important to note that all compatibility settings are optional in nature - applications can freely ignore all behaviors described within this section and these settings should not be added unless compatibility is specifically needed in one or more cases. The compatibility settings are provided for backward compatibility with documents created in legacy applications. As such, a number of the settings reference specific applications and specific versions of those applications. This is solely for backward compatibility reasons, and any of those settings are not intended for use by typical applications.

[*Note*: These settings can also be expressed using the generic compatSetting element defined in ISO/IEC 29500-1. *end note*]

[*Example*: Consider the following WordprocessingML fragment for the compatibility settings in a WordprocessingML document:

```
<w:settings>
  …
  <w:compat>
    <w:noTabHangInd />
  </w:compat>
</w:settings>
```

The compat element contains all of the document settings for this document. In this case, the single setting applied is the suppression of a tab stop when using a hanging indent using the noTabHangInd element (§14.7.3.31). *end example*]

14.7.3.1 alignTablesRowByRow (Align Table Rows Independently)

This element specifies whether applications shall align each row within a table independently based on the alignment setting of the jc element (Part 1, §17.4.28) when displaying the contents of a table in a WordprocessingML document.

When the justification of a table using the jc element is typically applied, that alignment is applied to the contents of the table (the table is centered, left justified, or right-aligned), and then individual rows are laid out based on the resulting table's position. This element, when present with a val attribute value of true (or equivalent), specifies that each table row shall be independently aligned based on the table alignment setting, ignoring the placement of all other rows.

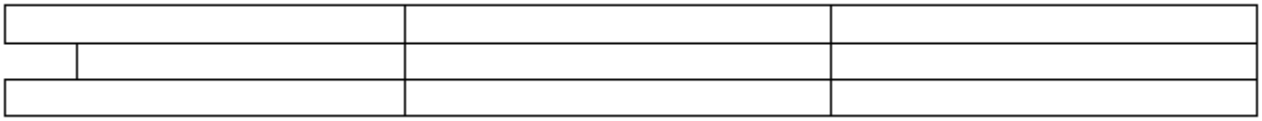
[*Example*: Consider a WordprocessingML document with a single centered table, whose second row is defined such that one-half of an inch is left before the row begins, as follows:

```
<w:tbl>
  <w:tblPr>
    <w:jc w:val="center" />
  </w:tblPr>
  <w:tr>
    …
  </w:tr>
  <w:tr>
    <w:trPr>
      <w:gridBefore w:val="1" />
```



```
        <w:wBefore w:w="720" w:type="dxa" />
      </w:trPr>
      …
    </w:tr>
    <w:tr>
      …
    </w:tr>
  </w:tbl>
```

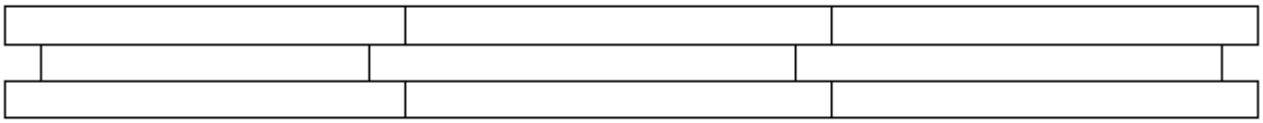
The default presentation would have the entire table centered, then the second row indented beyond that by 720 points:



However, if this compatibility setting is turned on:

```
<w:compat>
  <w:alignTablesRowByRow />
</w:compat>
```

Then that second row would instead be centered on the page independently of the other table rows, resulting in the following output:



In this case, the wBefore element's value is ignored, since the row was centered on the line as a row, and there is no table to be indented relative to. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.2 allowSpaceOfSameStyleInTable (Allow Contextual Spacing of Paragraphs in Tables)

This element specifies whether the suppression of additional space (contextual spacing) defined using the contextualSpacing element (Part 1, §17.3.1.9) shall be applied to paragraphs contained within tables.

Typically, the rules for the removal of additional paragraph spacing via the contextualSpacing element are applied to all paragraphs in a WordprocessingML document. This element, when present with a val attribute value of true (or equivalent), specifies that this setting shall always be ignored for paragraphs in table cells (and additional spacing shall be allowed).

[*Example*: Consider a WordprocessingML document with a default paragraph style with additional spacing after and contextual spacing set, as follows:

```
<w:style w:name="Normal" w:default="1">
  …
  <w:pPr>
    <w:spacing w:after="200" />
    <w:contextualSpacing />
  </w:pPr>
</w:style>
```

The default presentation would have the spacing suppressed between all paragraphs, since they are all of the default paragraph style defined above (contextual spacing applies):

On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks. When you create pictures, charts, or diagrams, they also coordinate with your current document look.

On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks. When you create pictures, charts, or diagrams, they also coordinate with your current document look.

You can easily change the formatting of selected text in the document text by choosing a look for the selected text from the Quick Styles gallery on the Home tab. You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly. On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks. When you create pictures, charts, or diagrams, they also coordinate with your current document look.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:allowSpaceOfSameStyleInTable />
</w:compat>
```

Then the paragraphs in the table never have their spacing suppressed, resulting in the following output:

On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks. When you create pictures, charts, or diagrams, they also coordinate with your current document look.
On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks. When you create pictures, charts, or diagrams, they also coordinate with your current document look.

You can easily change the formatting of selected text in the document text by choosing a look for the selected text from the Quick Styles gallery on the Home tab. You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly.

On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks. When you create pictures, charts, or diagrams, they also coordinate with your current document look.

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.3 autofitToFirstFixedWidthCell (Allow Table Columns To Exceed Preferred Widths of Constituent Cells)

This element specifies that when performing an AutoFit on a table in a WordprocessingML document in order to display it, applications shall alter that logic slightly in order to mimic the behavior of a previous word processing application.

Normally, the AutoFit behavior of a table is as is described in the associated simple type. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that this logic shall be changed as follows:

- If the width of a grid column in a table has been set by a preferred table cell width, then that column's width can be enlarged by the content of cells which themselves do not have a preferred width (in contrast, the normal logic never allows the content of cells to override a preferred width on a grid column).

[*Example*: Consider a WordprocessingML table with only one preferred cell width, a width of `720` points on the second cell in the first column, as follows:

```
<w:tbl>
  …
  <w:tr>
    <w:tc>
      <w:p/>
```



```
      </w:tc>
      <w:tc>
        <w:p/>
      </w:tc>
    </w:tr>
    <w:tr>
      <w:tc>
        <w:tcPr>
          <w:tcW w:w="720" w:type="dxa" />
        </w:tcPr>
        <w:p/>
      </w:tc>
      <w:tc>
        <w:p/>
      </w:tc>
    </w:tr>
  </w:tbl>
```

The default presentation would have the first column constrained to 720 points by the preferred width of the second cell in the first column:

This is an example of a cell with lots of content.	

However, if this compatibility setting is turned on:

```
  <w:compat>
    <w:autofitToFirstFixedWidthCell />
  </w:compat>
```

Then the column would be resized proportionally based on the content (ignoring the preferred width in that row), resulting in the following output:

This is an example of a cell with lots of content.	

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.4 autoSpaceLikeWord95 (Incorrectly Adjust Text Spacing for Specific Unicode Ranges)

This element specifies adjustments (detailed below) which should be applied to the spacing between adjoining regions of non-ideographic and ideographic text when the autoSpaceDE (Part 1, §17.3.1.2) and autoSpaceDN (Part 1, §17.3.1.3) elements have a value of `true` (or equivalent). This algorithm typically results in the following:

- An increase in the inter-character spacing added between non-ideographic and/or number characters and certain full-width characters
- No inter-character spacing between non-ideographic and/or number characters and certain half-width characters

Typically, applications apply additional spacing between ideographic and non-ideographic characters/numeric characters when the autoSpaceDE / autoSpaceDN properties are applied. This element, when present with a val attribute value of `true` (or equivalent), specifies that applications shall apply the following adjustments to this logic:

- Characters in the following Unicode ranges should be treated as ideographic, even though those characters are full-width forms of non-ideographic text: U+FF10–U+FF19, U+FF21–U+FF3A, and U+FF41–U+FF5A. [*Note*: This results in the unnecessary addition of space. *end note*]
- Characters in the following Unicode ranges should be treated as non-ideographic, even though those characters are ideographic: U+FF66–U+FF9F. [*Note*: This results in the omission of the intended additional space. *end note*]

[*Example*: Consider a WordprocessingML document with two paragraphs containing a mix of East Asian and Latin characters:

```
<w:p>
  <w:r>
    <w:t>ab</w:t>
  </w:r>
  <w:r>
    <w:t>ｱ</w:t>
  </w:r>
  <w:r>
    <w:t>ｱ</w:t>
  </w:r>
  <w:r>
    <w:t>cd</w:t>
  </w:r>
</w:p>
<w:p>
  <w:r>
```



```
      <w:t>ab</w:t>
    </w:r>
    <w:r>
      <w:t>２</w:t>
    </w:r>
    <w:r>
      <w:t>２</w:t>
    </w:r>
    <w:r>
      <w:t>cd</w:t>
    </w:r>
  </w:p>
```

The first paragraph contains characters with Unicode value U+FF66 (ｦ). The second paragraph contains characters with Unicode value U+FF12 (２). If autoSpaceDE is `true`, spacing is added in the first paragraph (between the ideographs and the non-ideographic characters), but not in the second (all four characters are not ideographs):

ab ｦｦ cd

ab ２ ２ cd

If this compatibility setting is turned on:

```
  <w:compat>
    <w:autoSpaceLikeWord95 />
  </w:compat>
```

Then, although it appears incorrect, applications should not add space in the first paragraph and should apply it in the second:

abｦｦcd

ab ２ ２ cd

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.5 cachedColBalance (Use Cached Paragraph Information for Column Balancing)

This element specifies whether applications shall incorrectly calculate the height of a paragraph for the purposes of column balancing when rendering WordprocessingML documents. Specifically, this element specifies that when a paragraph's lines have differing heights, an application shall treat this paragraph as though it had only one line equaling the full paragraph height, regardless of the actual number of lines in the paragraph.

[*Guidance*: It is recommended that applications not intentionally replicate this behavior; it is maintained only for compatibility with existing documents from a legacy application. *end guidance*]

Typically, lines are correctly measured for their height when balancing columns as part of a WordprocessingML document. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that applications shall perform the incorrect calculation in the conditions described above.

[*Example*: Consider a WordprocessingML document with two columns of text which shall be balanced.

If this compatibility setting is turned on:

```
<w:compat>
  <w:cachedColBalance />
</w:compat>
```

Then applications should perform the calculation described above to balance the columns, as needed. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.6 convMailMergeEsc (Treat Backslash Quotation Delimiter as Two Quotation Marks)

This element specifies whether applications should perform a conversion of the contents of a mail merge data source when reading those contents in order to perform a mail merge operation with their contents.

Typically, the contents of a mail merge data source are read in exactly as specified when performing a mail merge with the contents of a data source. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that applications shall interpret delimiters composed of a backslash and quotation mark (\") as two quotation marks ("), within external data sources to be connected to via a mail merge.

[*Example*: Consider a WordprocessingML document with the following content in its data source:

```
This is a \"test\".
```

The default presentation would have the resulting merged data read in just as it appears:

```
This is a \"test\".
```

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:convMailMergeEsc />
</w:compat>
```

Then instances of a backslash and quotation mark would be converted, resulting in the following output:

```
This is a ""test"".
```

end example]

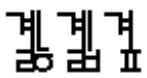
This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.7 displayHangulFixedWidth (Always Use Fixed Width for Hangul Characters)

This element specifies whether applications should assume that all characters in the Hangul Syllables Unicode sub range (character values between 0xAC00 and 0xD7FF) are of a single fixed width or shall use the characters widths defined by the font in use (typical for a proportional width font).

Typically, applications shall retrieve the character width for any character in a document from the associated font, allowing each character to be of its own width (a proportional width character). This element, when present with a `val` attribute value of `true` (or equivalent), specifies that applications shall instead assume a single fixed width for all characters in the Hangul Syllables sub range, by reading the width of Unicode character 0x4E00 from the associated font and using that width for all Hangul characters (or, if that character is not present, the next available character in the font).

[*Example*: Consider a WordprocessingML document with three Hangul characters:



The default presentation would have each of those characters using the widths defined by the font (the highlighting indicates that each character has its own width):

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:displayHangulFixedWidth />
</w:compat>
```

Then all three characters are forced to the fixed width of character 0x4E00 from the font (or, in this case, the next available character), resulting in the characters in the font being forced to that fixed width, which results in the following output:

Notice from the highlighting that the characters have been compressed to the width of the single character and displayed at that fixed width. *end example*]

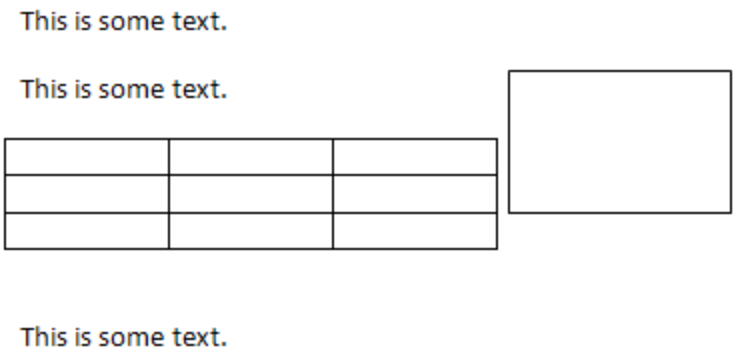
This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.8 doNotAutofitConstrainedTables (Do Not AutoFit Tables To Fit Next To Wrapped Objects)

This element specifies whether applications shall allow tables to be resized to the remaining available line width when they are using the AutoFit algorithm and part of that line is filled by a shape with a wrapping type with a value of `square` or `tight`.

Typically, a table which is AutoFit and has a preferred width shall have its width reduced in order to allow a floating shape to wrap around its contents within the document, as that shape simply reduces the width of the line and the AutoFit algorithm applies to the remaining line width. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that tables shall never have any preferred width overridden to allow them to wrap around that floating object, and shall instead be pushed to the next full width line in the document to be displayed.

[*Example*: Consider a WordprocessingML document with a floating shape centered in the document, followed by a table with preferred cell widths of 2.22", as follows:

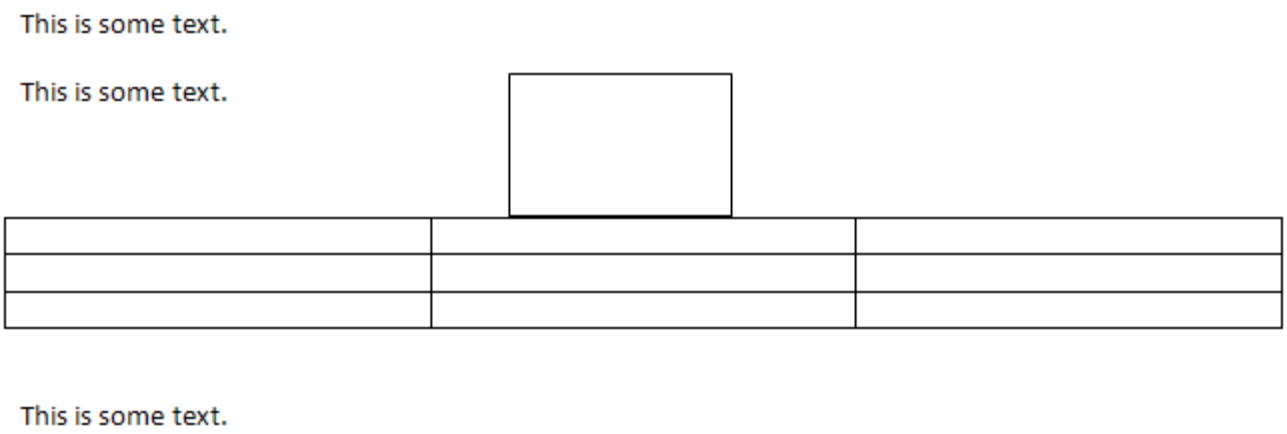


The default presentation of this document overrides the preferred cell widths to force the table to fit on the line next to the floating shape with tight wrapping.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:doNotAutofitConstrainedTables />
</w:compat>
```

Then that table is not resized, so it cannot fit and must be pushed to the next full width line, resulting in the following output:



end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

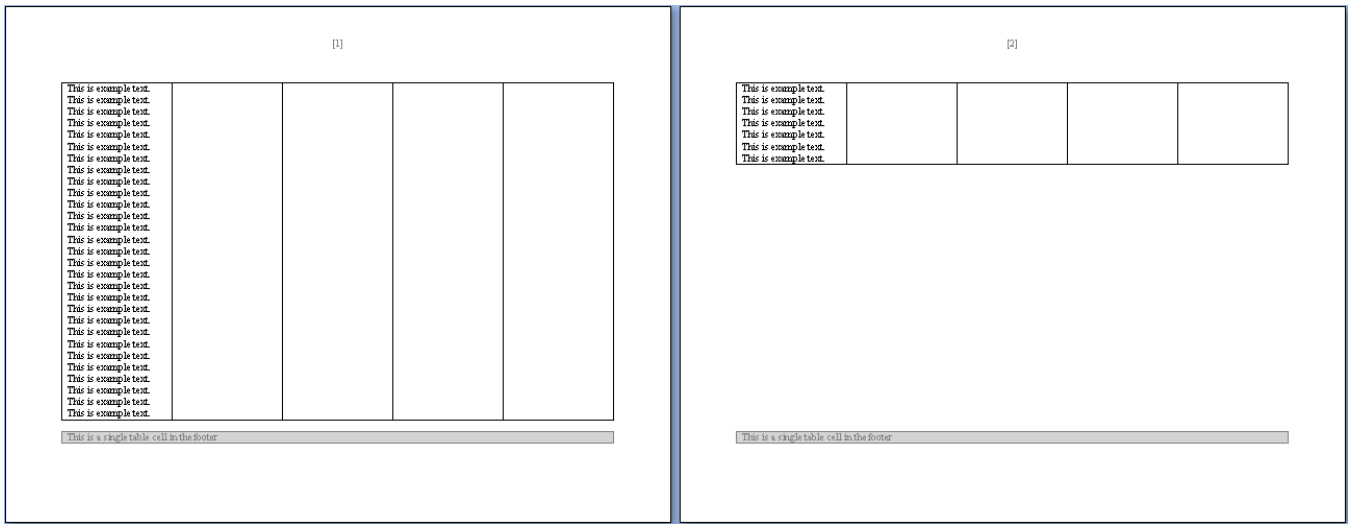
14.7.3.9 doNotBreakConstrainedForcedTable (Don't Break Table Rows Around Floating Tables)

This element specifies whether applications shall allow a table row to be split in two when its contents are displayed under the following circumstances:

- The table row exceeds one page in height (it shall be split into two pages)
- The table row would need to be split in order to accommodate a floating table also on the page (tables which have been set to floating using the tblpPr element (Part 1, §17.4.58))

Typically, assuming the cantSplit property (Part 1, §17.4.6) is not set, a table row which cannot fit on one single page shall be split as needed around any floating table on a page, in order to allow its contents to be fully displayed across two or more pages. This element, when present with a val attribute value of `true` (or equivalent), specifies that table rows which exceed one page in height shall never be split around floating tables in the document, and shall instead be displayed on the first page below the floating table, even if that means that part of the table row is clipped by the edge of the page.

[*Example*: Consider a WordprocessingML document with a long single table row which must be split across two separate pages in the document, in order to accommodate a floating table anchored in the footer, as follows:

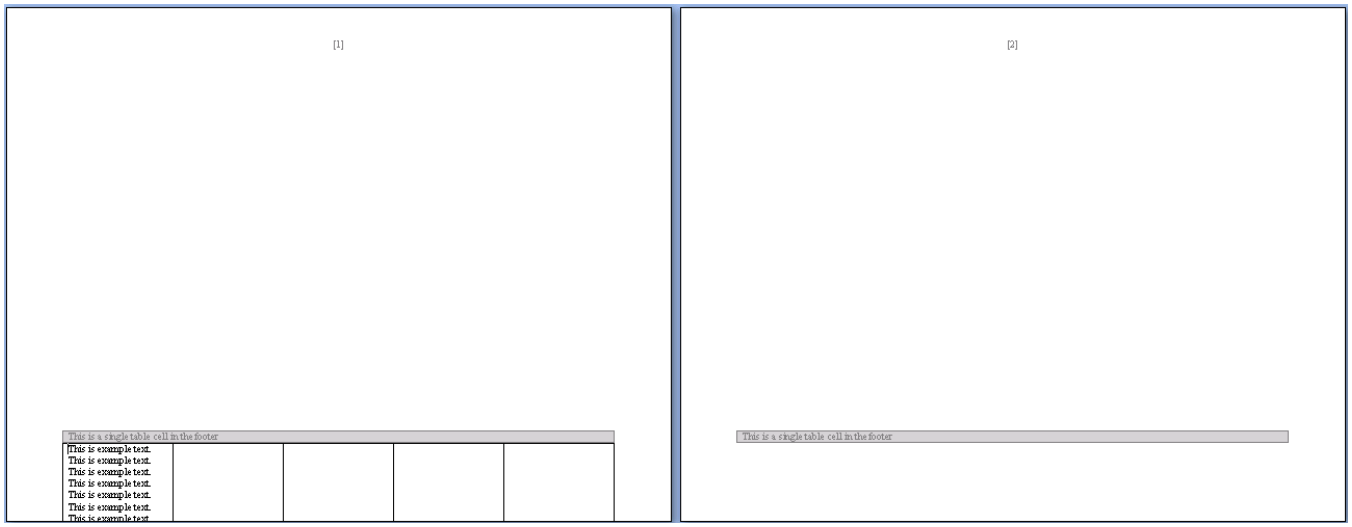


The default presentation of this document forces that row to be split as needed around that floating table.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:doNotBreakConstrainedForcedTable />
</w:compat>
```

Then that table row is never split around the floating table, so it is always placed below that floating table on the page, and allowed to flow off the page as needed, resulting in the following output:



This example, while extreme, shows how the row is placed below the floating table, rather than breaking around it. *end example*]

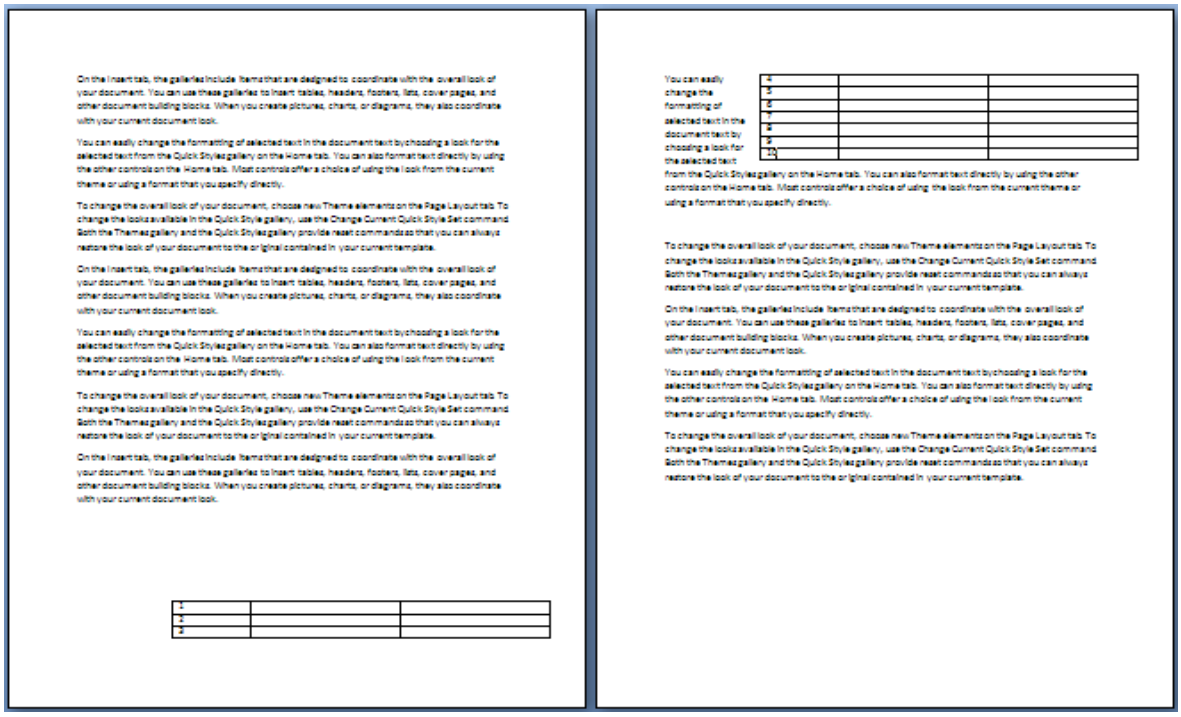
This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.10 doNotBreakWrappedTables (Do Not Allow Floating Tables To Break Across Pages)

This element specifies whether applications shall allow tables which have been set to floating using the tblpPr element (Part 1, §17.4.58) shall be allowed to break across multiple pages when needed.

Typically, a table whose contents cannot all be displayed on one page is broken as needed across multiple pages in order to preserve the location of the table (just as a paragraph of multiple lines is broken across pages as needed). This element, when present with a val attribute value of `true` (or equivalent), specifies that floating tables shall never be broken across pages, and shall instead be put on the first page by adjusting the starting position of the table as needed to fit on that single page.

[*Example*: Consider a WordprocessingML document with a floating table positioned at the bottom of a page , as follows:

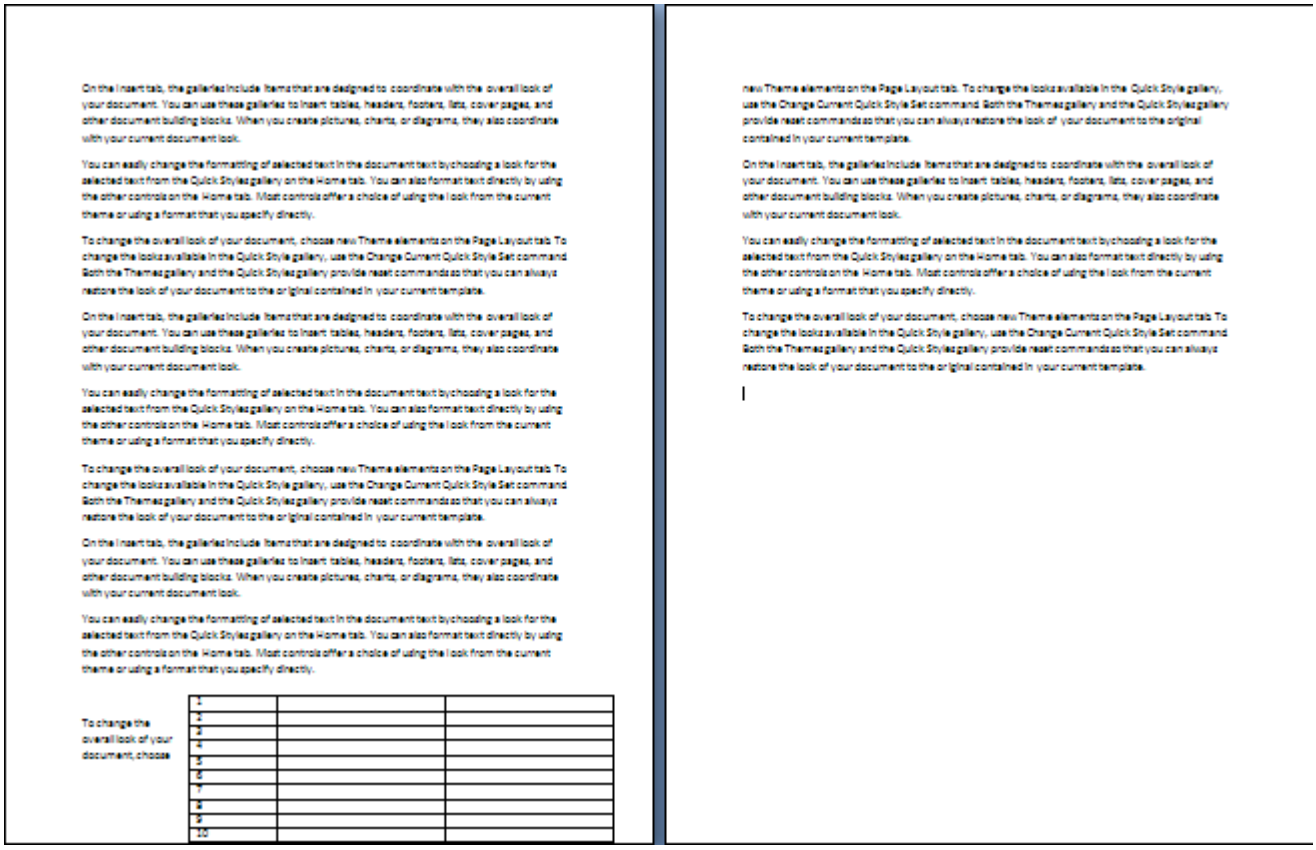


The default presentation of this document results in that table being broken across two pages of content.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:doNotBreakWrappedTables />
</w:compat>
```

Then that table is not broken across the page boundary, so it must be moved further up on the first page to accommodate its entire size, resulting in the following output:



Notice that the table now flows into the page margins in order to keep it on one page. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.11 doNotSnapToGridInCell (Do Not Snap to Document Grid in Table Cells with Objects)

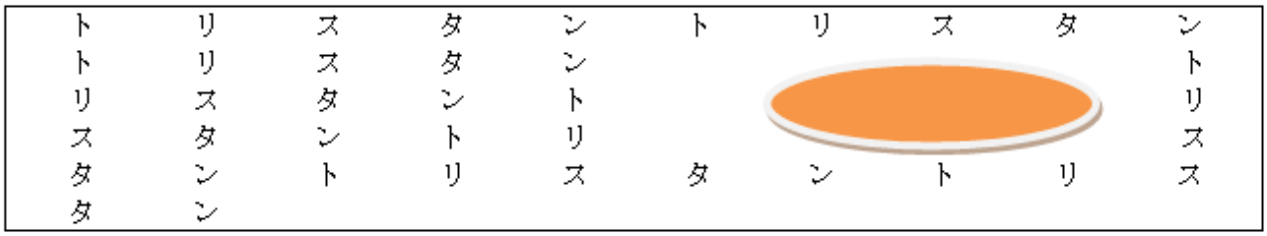
This element specifies whether a document grid defined using the docGrid element (Part 1, §17.6.5) shall be applied to the contents of table cells in that section which also contain floating objects. Note that the floating object shall be part of the cell, and simply not displayed over the cell due to its anchoring relative to another part of the document.

Typically, if a floating object is present in a table cell, then that setting shall have no impact on whether East Asian text in that cell is snapped to the document grid (as text is always snapped to the grid). This element, when present with a val attribute value of `true` (or equivalent), specifies that whenever a floating object is present in a table cell, that the cell's contents shall not be snapped to the document grid.

[*Example*: Consider a WordprocessingML document consisting of a single section, whose document grid settings specify that each page must be exactly 10 characters wide, as follows:

```
<w:sectPr>
  <w:docGrid w:type="snapToChars" w:charSpace="146636" />
</w:sectPr>
```


If this document contains a table with a single cell, containing some text and a single floating shape, the contents of the cell are still snapped to the 10 characters per line character grid, as follows:



However, if this compatibility setting is turned on:

```
<w:compat>
  <w:doNotSnapToGridInCell />
</w:compat>
```

Then the presence of a floating object in each cell must result in the document grid setting being ignored, resulting in the following output:

The additional character pitch was still added to each character on the line, but those characters are no longer snapped to the document grid. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.12 doNotSuppressIndentation (Do Not Ignore Floating Objects When Calculating Paragraph Indentation)

This element specifies whether applications should ignore the presence of floating objects when calculating the starting position of paragraphs which are wrapped around floating objects.

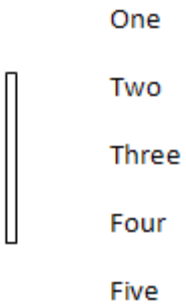
Typically, the presence of a floating object on the same line or lines as a paragraph shall only affect the text when the floating object occurs where that text would normally be presented. [*Example*: Text at a 1" indentation would only be displaced by a floating object that appears at that position and not one that appears from 0" to 0.5" on the same line. *end example*].

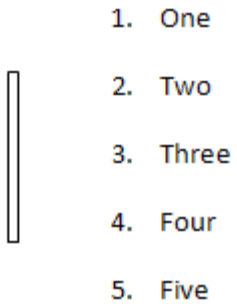
This element, when present with a val attribute value of `true` (or equivalent), specifies that floating objects shall always impact paragraphs on the same line in two ways:

- If the paragraph is not numbered, then it shall tightly wrap any floating object which precedes it on the same line, ignoring its own indentation settings. [*Example*: A paragraph with a 1" left indent shall tightly wrap a floating object which appears at only 0.25" on the same line. *end example*]
- If the paragraph is numbered using the numPr element (Part 1, §17.3.1.19), then it shall calculate and use its full indent relative to the edge of the floating object, not relative to the edge of the page. [*Example*: A numbered paragraph with a 1" left indent must appear 1.5" into the page if it is preceded by a floating object which appears at 0.5" on the same line. *end example*]

[*Example*: Consider a WordprocessingML document with a narrow floating object at 0.5" on the page, surrounded by both numbered and unnumbered paragraphs.

The default presentation would have no impact on the paragraphs based on that floating object, since the two do not intersect:





However, if this compatibility setting is turned on:

```
<w:compat>
  <w:doNotSuppressIndentation />
</w:compat>
```

Then the two alternate rules defined above would apply, resulting in the following output:

One

Two

Three

Four

Five

1. One
2. Two
3. Three
4. Four
5. Five

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.13 doNotSuppressParagraphBorders (Do Not Suppress Paragraph Borders Next To Frames)

This element specifies whether applications should suppress paragraph borders defined using the pBdr element (Part 1, §17.3.1.24) when those borders would be displayed next to the contents of paragraphs which have been defined as frames using the framePr element (Part 1, §17.3.1.11).

Typically, when a paragraph's borders appear next to a frame, those borders are suppressed to avoid having two borders in close proximity. This element, when present with a val attribute value of `true` (or equivalent), specifies that those borders shall not be suppressed.

[*Example*: Consider a WordprocessingML document with a paragraph with a paragraph border that is bounded on its bottom let side by a text frame.

The default presentation would suppress the borders which intersect the frame (in this case, the right border of lines three through eight):

Example text. Example text. Example text. Example text. Example text. Example text. Example text.
Example text. Example text. Example text. Example text. Example text. Example text. Example text.

This is a frame.

Example text. Example text. Example text. Example text. Example text.
Example text. Example text. Example text. Example text. Example text.
Example text. Example text. Example text. Example text. Example text.
Example text. Example text. Example text. Example text. Example text.
Example text. Example text. Example text. Example text. Example text.
Example text. Example text. Example text.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:doNotSuppressParagraphBorders />
</w:compat>
```

Then no border suppression must take place, resulting in the following output:

Example text. Example text. Example text. Example text. Example text. Example text. Example text.
Example text. Example text. Example text. Example text. Example text. Example text. Example text.

This is a frame.

Example text. Example text. Example text. Example text. Example text.
Example text. Example text. Example text. Example text. Example text.
Example text. Example text. Example text. Example text. Example text.
Example text. Example text. Example text. Example text. Example text.
Example text. Example text. Example text. Example text. Example text.
Example text. Example text. Example text.

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.14 doNotUseEastAsianBreakRules (Do Not Compress Compressible Characters When Using Document Grid)

This element specifies whether applications should compress characters with identical compression rules when the document grid has been defined using the docGrid element (Part 1, §17.6.5). *Compression rules* refer to the additional bearing on the left and/or right side of a typical character, which can be compressed as needed without modifying the actual width of the character (its breadth).

Typically, punctuation characters with an identical set of compression rules are compressed when the contents of a document are displayed. This element, when present with a val attribute value of `true` (or equivalent), specifies that if a document grid is defined for the current section, compression shall never be performed on any character - all compressible characters shall be individually snapped to the document grid.

[*Example*: Consider a WordprocessingML document with a document grid set to allow 10 characters per line:

```
<w:sectPr>
  <w:docGrid w:type="snapToChars" w:charSpace="146636" … />
</w:sectPr>
```

The default presentation would allow characters with identical compression rules to compress and utilize a single slot on the document grid (notice that the four parenthesis on the first line are combined since they can be compressed identically, while the two parenthesis with different compression on line two are not):

```
あ    あ    あ    あ    ) ) ) )    v    あ    あ    あ    あ
あ    あ    あ    あ    a    あ    あ    )    (
```

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:doNotUseEastAsianBreakRules />
</w:compat>
```

Then no character with compression is compressed and instead are snapped to the grid individually, resulting in the following output:

```
あ    あ    あ    あ    )    )    )    )    v    あ
あ    あ    あ    あ    あ    あ    あ    a    あ    あ    )
(
```

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.15 doNotUseHTMLParagraphAutoSpacing (Use Fixed Paragraph Spacing for HTML Auto Setting)

This element specifies whether applications should use a fixed definition when interpreting automatic paragraph spacing defined by a value of `true` (or equivalent) on the beforeAutospacing and/or afterAutospacing attributes on the spacing element (Part 1, §17.3.1.33).

Typically, applications shall interpret these settings to match the behavior of most HTML user agents, mimicking the default spacing above and below an HTML p element without additional spacing information. This element, when present with a val attribute value of `true` (or equivalent), specifies that those two attributes shall result in the following settings for each value:

- beforeAutospacing = 5 points of spacing before
- afterAutospacing = 10 points of spacing after

[*Example*: Consider a WordprocessingML document with a three paragraphs using HTML autospacing, as follows:

```
<w:p>
  <w:pPr>
    <w:spacing w:beforeAutospacing="true" w:afterAutospacing="true" />
  </w:pPr>
  <w:r>
    <w:t>Paragraph One</w:t>
  </w:r>
</w:p>
<w:p>
  <w:pPr>
    <w:spacing w:beforeAutospacing="true" w:afterAutospacing="true" />
  </w:pPr>
  <w:r>
    <w:t>Paragraph Two</w:t>
  </w:r>
</w:p>
<w:p>
  <w:pPr>
    <w:spacing w:beforeAutospacing="true" w:afterAutospacing="true" />
  </w:pPr>
  <w:r>
    <w:t>Paragraph Three</w:t>
  </w:r>
</w:p>
```

The default presentation would result in output designed to match that of all common HTML user agents:

Paragraph One.

Paragraph Two.

Paragraph Three.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:doNotUseHTMLParagraphAutoSpacing />
</w:compat>
```


Then the paragraphs has exact spacing of 5 points before and 10 points after, resulting in the following output:

Paragraph One.

Paragraph Two.

Paragraph Three.

Notice that the paragraphs are more condensed in the second example. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.16 doNotUseIndentAsNumberingTabStop (Ignore Hanging Indent When Creating Tab Stop After Numbering)

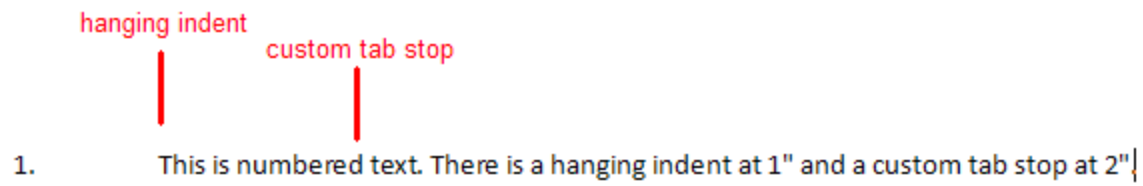
This element specifies whether applications shall use the custom tab stop generated by the hanging indent (if any) when advancing the text after the numbering for a numbered paragraph.

Typically, a hanging indent on a paragraph creates a virtual custom tab stop at that location, and therefore a tab added after the numbering on a numbered paragraph by the suff element (Part 1, §17.9.29) shall advance to that tab stop, so that the text of the numbered paragraph begins at that location. This element, when present with a val attribute value of `true` (or equivalent), specifies that a tab stop added as the suffix to the numbering of a numbered paragraph shall ignore that virtual custom tab stop and shall instead advance to the next real tab stop (custom or automatic) on the current line.

[*Example*: Consider a WordprocessingML document with numbering, whose first level of numbering specifies a tab stop suffix, a hanging indent at 1", and a custom tab stop at 2":

```
<w:abstractNum w:abstractNumId="0">
  …
  <w:lvl w:ilvl="0">
    <w:suff w:val="tab" />
    <w:pPr>
      <w:ind w:left="1440" w:hanging="1440" />
      <w:tabs>
        <w:tab w:val="2880" />
      </w:tabs>
    </w:pPr>
  </w:lvl>
</w:abstractNum>
```

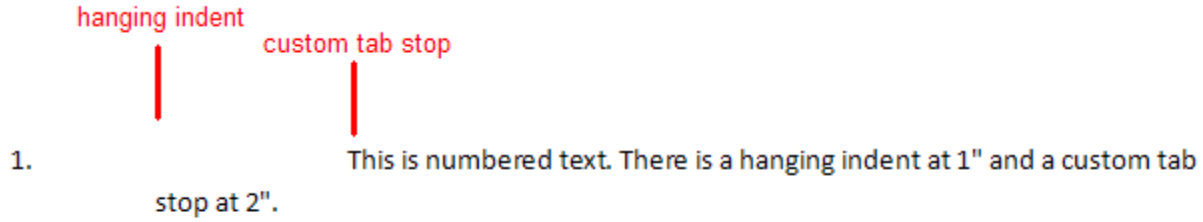
The default presentation of this document results in the tab stop generated by the numbering advancing to the virtual tab stop generated by the hanging indent at 1", as follows:



However, if this compatibility setting is turned on:

```
<w:compat>
  <w:doNotUseIndentAsNumberingTabStop />
</w:compat>
```

Then that tab suffix ignores the virtual tab stop of the hanging indent, so it must advance to the next custom tab stop on the line (at 2"), resulting in the following output:



end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

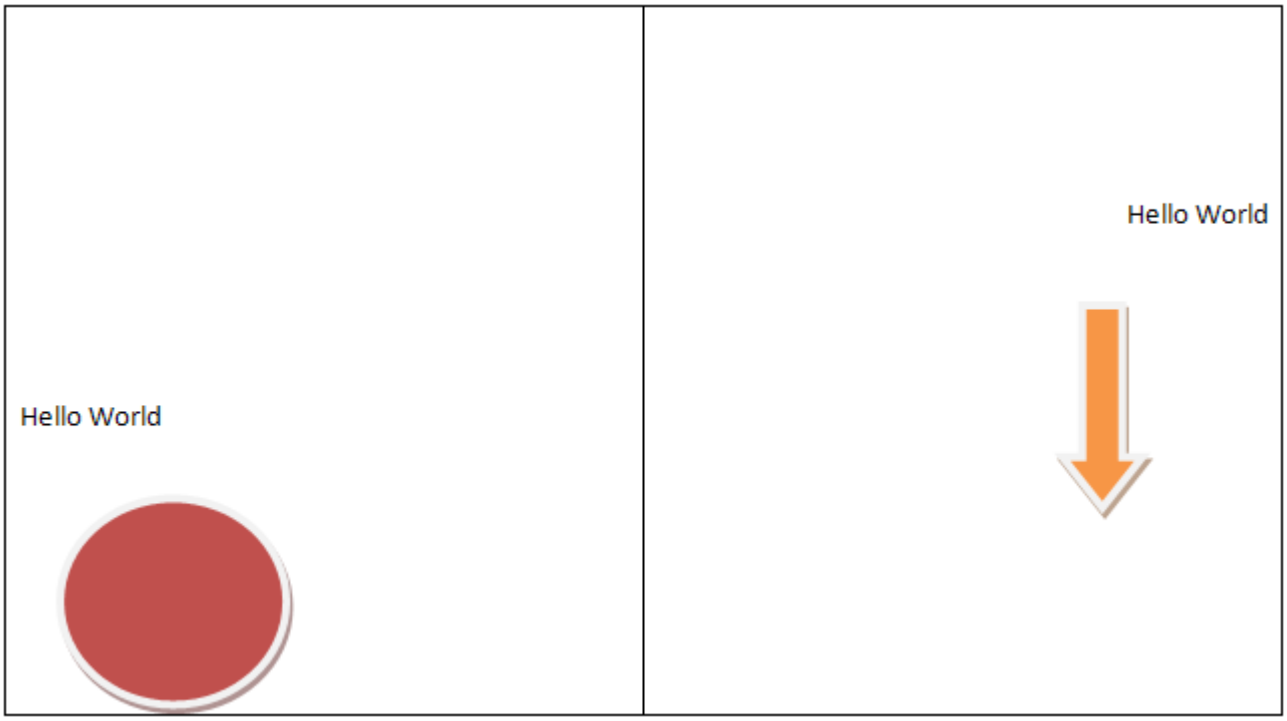
14.7.3.17 doNotVertAlignCellWithSp (Don't Vertically Align Cells Containing Floating Objects)

This element specifies whether applications shall vertically align the contents of a table cell, even when the contents of that table cell include one or more floating objects. Note that the floating object shall be part of the cell, and simply not displayed over the cell due to its anchoring relative to another part of the document.

Typically, if the alignment of a table cell in a WordprocessingML document is specified, then the entire contents of that cell are aligned as specified [*Example*: The entire contents of the cell are centered vertically and moved right-aligned horizontally at that point. *end example*]. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that whenever a floating object is present in a table cell, that no vertical alignment shall be applied to the contents of that cell, and the contents of the cell shall instead always be top aligned to the cell's contents.

[*Example*: Consider a WordprocessingML table with two cells, each containing some text and a single floating shape. The first cell is vertically aligned to the bottom of the cell, and the second cell is vertically aligned to the center of the cell.

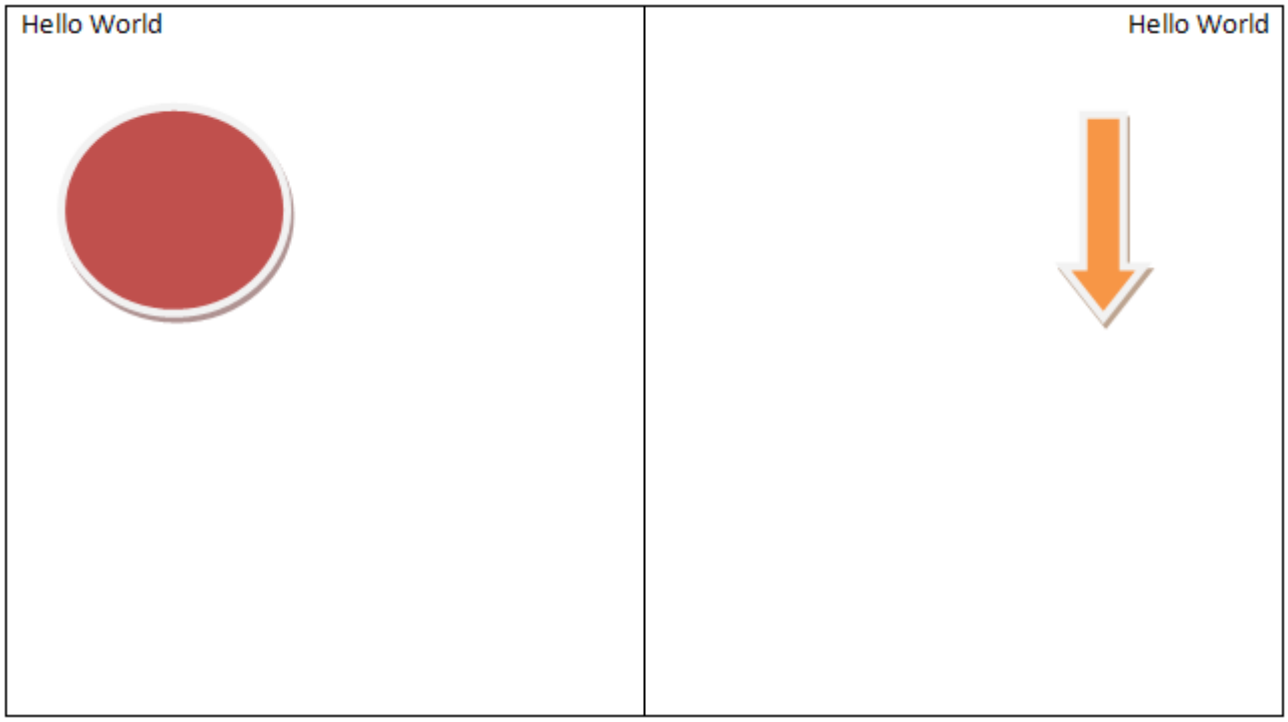
The default presentation of this document results in each cell (including the extents of the floating objects) being vertically aligned as specified, as follows:



However, if this compatibility setting is turned on:

```
<w:compat>
  <w:doNotVertAlignCellWithSp />
</w:compat>
```

Then the presence of a floating object in each cell must result in the vertical alignment setting being ignored (each vertical alignment must be top-aligned relative to the cell), resulting in the following output:



end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.18 doNotVertAlignInTxbx (Ignore Vertical Alignment in Textboxes)

This element specifies whether applications shall allow text within text boxes to be vertically aligned when the `v-text-anchor` property is set within the parent VML shape.

Typically, if when the `v-text-anchor` property is set within the parent VML shape, then based on the value of that property, the text is top, center, or bottom aligned appropriately. This element, when present with a val attribute value of `true` (or equivalent), specifies that the property shall be ignored, and instead the contents of the table shall always be top-aligned.

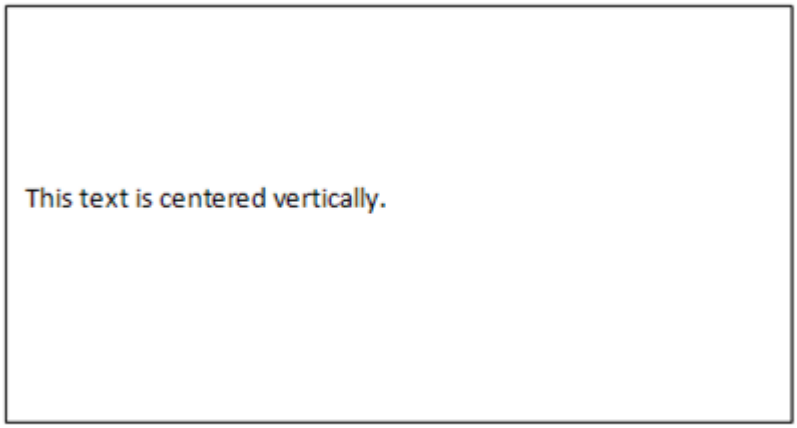
[*Example*: Consider a WordprocessingML table with a single center-aligned text box:

```
<v:shape id="_x0000_s1026" type="#_x0000_t202" style="v-text-anchor:middle">
  <v:textbox>
    <w:txbxContent>
      <w:p>
        <w:r>
          <w:t>This text is centered vertically.</w:t>
        </w:r>
      </w:p>
```



```
      </w:txbxContent>
    </v:textbox>
  </v:shape>
```

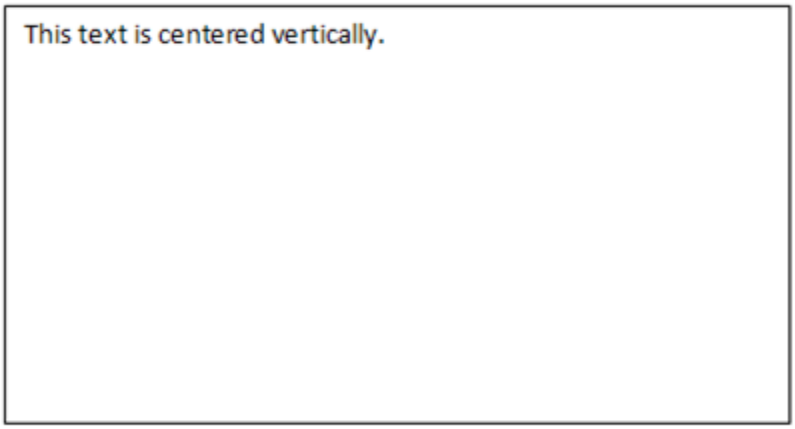
The default presentation of this document results in the contents of the text box being center aligned, as follows:



However, if this compatibility setting is turned on:

```
  <w:compat>
    <w:doNotVertAlignInTxbx />
  </w:compat>
```

Then the text must always be top aligned, regardless of the `-text-anchor` property, resulting in the following output:



end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.19 doNotWrapTextWithPunct (Do Not Allow Hanging Punctuation With Character Grid)

This element specifies whether applications shall allowing hanging punctuation when:

- The overflowPunct element (Part 1, §17.3.1.21) is turned on for a paragraph
- A document grid is defined using the docGrid element (Part 1, §17.6.5) which defines the number of characters per line

Typically, paragraphs which allow hanging punctuation shall allow the number of characters on a line as specified by the document grid to be exceeded by one in order to allow for hanging punctuation. This element, when present with a val attribute value of `true` (or equivalent), specifies that the document grid shall never be exceeded for hanging punctuation.

[*Example*: Consider a WordprocessingML document with a document grid set to allow 10 characters per line:

```
<w:sectPr>
  <w:docGrid w:type="snapToChars" w:charSpace="146636" … />
</w:sectPr>
```

If the eleventh character on the line was a punctuation characters, the default presentation would allow that character to behave as hanging punctuation on the first line:

“ 言 葉 が 言 葉 が 言 葉 が ”

言 葉 が 言 葉

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:doNotwrapTextWithPunct />
</w:compat>
```

Then the character grid cannot be exceeded even for the hanging punctuation, resulting in the following output:

“ 言 葉 が 言 葉 が 言 葉

が ” 言 葉 が 言 葉

The hanging punctuation was disallowed, moving it (and the character before it, since that character cannot begin a line) to the following line. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.20 footnoteLayoutLikeWW8 (Ignore Page Break from Continuous Section Break)

This element specifies that applications should override the default behaviour for a continuous section break when one or more footnotes are present on the page with the footnote. This override typically results in text being displayed on the same page as a continuous section break (after the break, which would normally move all following text to the next page).

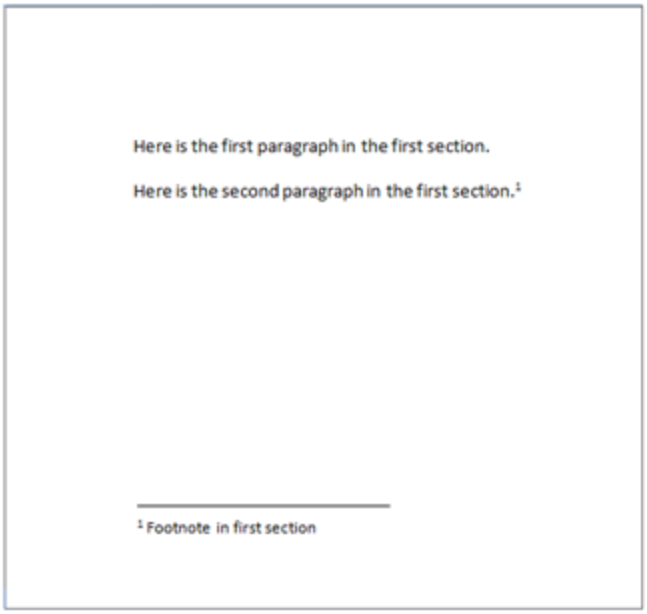
Typically, applications render a continuous section break as a page break when one or more footnoteRef elements (Part 1, §17.11.13) occur on that page before the break, as described in Part 1, §17.18.77. This element, when present with a val attribute value of `true` (or equivalent), specifies that applications should allow any paragraph after the section break that contains no footnoteRef elements (Part 1, §17.11.13) to be displayed on the same page. If the resulting content reaches the page extents, the section's page break is ignored.

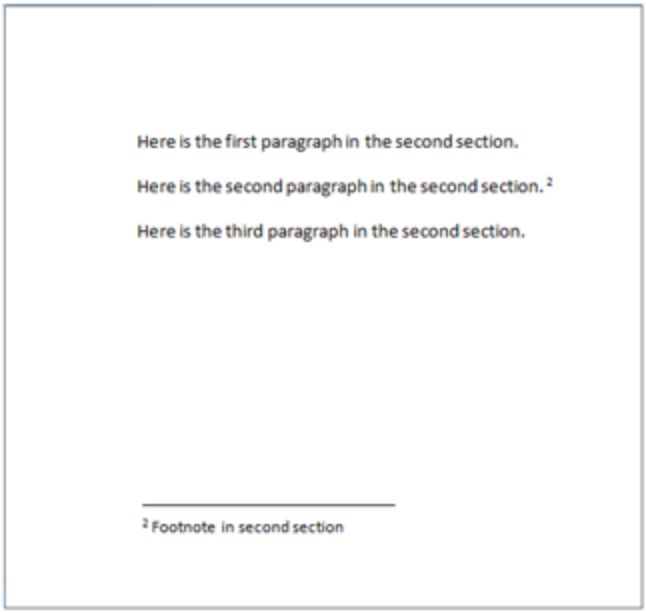
[*Example*: Consider a WordprocessingML document with two footnotes contained in two sections, separated by a continuous section break:

```
<w:p>
  <w:r>
    <w:t xml:space="preserve">Here is the first paragraph in the first
section.</w:t>
  </w:r>
</w:p>
<w:p>
  <w:r>
    <w:t>Here is the second paragraph in the first section.</w:t>
  </w:r>
  <w:r>
    <w:rPr>
      <w:rStyle w:val="FootnoteReference" />
    </w:rPr>
   <w:footnoteReference w:id="2" />
    </w:r>
</w:p>
<w:p/>
<w:p>
  <w:pPr>
    <w:sectPr>
      …
    </w:sectPr>
  </w:pPr>
```

```
    </w:p>
  <w:p>
    <w:r>
      <w:t>Here is the first paragraph in the second section.</w:t>
    </w:r>
  </w:p>
  <w:p>
    <w:r>
      <w:t xml:space="preserve">Here is the second paragraph in the second
  section.</w:t>
    </w:r>
    <w:r>
      <w:rPr>
        <w:rStyle w:val="FootnoteReference" />
        </w:rPr>
      <w:footnoteReference w:id="3" />
    </w:r>
  </w:p>
  <w:p>
    <w:r>
      <w:t xml:space="preserve">Here is the third paragraph in the second section.
  </w:t>
      </w:r>
  </w:p>
  <w:sectPr>
    <w:type w:val="continuous" />
      …
  </w:sectPr>
```

The default rendering of such a document results in the continuous section break as a page break, resulting in the following two page document:

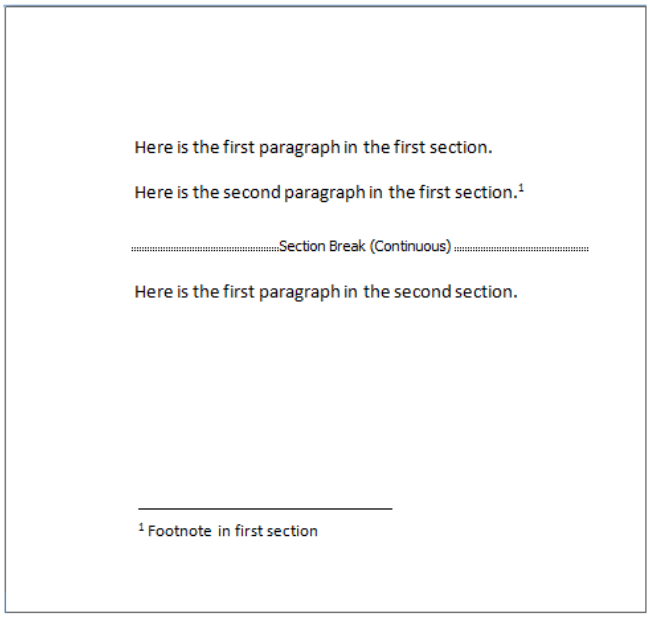


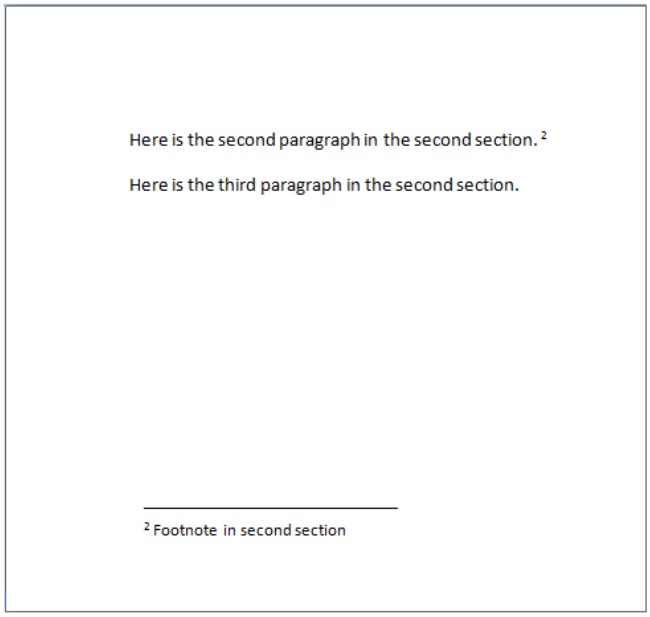


However, if this compatibility setting is turned on:

```
<w:compat>
  <w:footnoteLayoutLikeWW8 />
</w:compat>
```

Then the first paragraph following the section break (not having any footnote references) is displayed on the same page, despite the section break, resulting in the following output:





end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.21 forgetLastTabAlignment (Ignore Width of Last Tab Stop When Aligning Paragraph If It Is Not Left Aligned)

This element specifies how applications should handle the final tab stop on a line when aligning the contents of a paragraph as specified by the jc element (Part 1, §17.3.1.13) in the paragraph's properties.

Typically, aligning the contents of a paragraph involves the following:

- Determining the layout of that line before the alignment (including all tab stops)
- Aligning the resulting contents of the line

This is done to ensure that tab stops on a line do not change when the contents of the paragraph are aligned (i.e. the tab stops should not have to take into account the paragraph alignment).

This element, when present with a val attribute value of `true` (or equivalent), specifies that applications shall ignore the additional line width generated by the last tab stop (and only the last tab stop) when the alignment of the tab stop as defined by the val attribute on the tab element (Part 1, §17.3.1.37) is not `left` (or `bar`, which as defined by ISO/IEC 29500, is not a tab stop per se) when determining the width of the line. The resulting full line shall then be aligned at the position where the line would have been aligned without that tab stop.

[*Example*: Consider a WordprocessingML document with two center aligned paragraphs of text - the first also containing a centered tab stop positioned at 2":


```
<w:p>
  <w:pPr>
    <w:tabs>
      <w:tab w:val="center" w:pos="2880" />
    </w:tabs>
    <w:jc w:val="center" />
  </w:pPr>
  <w:r>
    <w:t>Text Before</w:t>
    <w:tab/>
    <w:t>Text After</w:t>
  </w:r>
</w:p>
<w:p>
  <w:pPr>
    <w:jc w:val="center" />
  </w:pPr>
  <w:r>
    <w:t>Text BeforeText After</w:t>
  </w:r>
</w:p>
```

The default presentation would determine the full width of each line including the tab stops, finally aligning the resulting text to the `center` position as requested by the jc element:

Text Before Text After

Text BeforeText After

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:forgetLastTabAlignment />
</w:compat>
```

Then the width added to the line by the last tab is ignored when centering the paragraph because that tab is a `center` aligned tab stop, resulting in the following output:

Text Before Text After

Text BeforeText After

In the resulting output, the starting location of both lines is at the same place on the page, as the resulting width of both lines is identical when the tab stop is removed from the line width calculation. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.22 growAutofit (Allow Tables to AutoFit Into Page Margins)

This element specifies whether applications shall allow a table which is using the AutoFit table layout algorithm to extend beyond the margins of the page if the minimum width of each table cell would result in an overall table width which is wider than those page margins.

Typically, if a table is using the AutoFit layout algorithm, then based on the definition of that logic, each column in the table shall be increased to the minimum width of its contents (e.g. the longest non-breaking run of text contained within it and/or the width of an inline image contained in one of its cells) until the overall width of the table reaches that of the text extents on the page, at which point text shall be broken and images shall be clipped as needed to maintain the width of the table at the page width (i.e. the page width is an immutable maximum width for the table). This element, when present with a `val` attribute value of `true` (or equivalent), specifies that the minimum width of the cells shall not be constrained by the page width, and instead the table shall be allowed to extend into the page margins as needed in order to meet the minimum widths of each of its cells.

[*Example*: Consider a WordprocessingML table with three cells in each row. If the contents of each cell in that first row each contain a long non-breaking string (such that the minimum widths of each cell's contents exceed the page width), then the rules for table AutoFit specify that each cell must be broken proportionally when the overall width of the table reaches the page width.

The default presentation of this document results in each cell being broken as needed to maintain the table width, as follows:

veryverylongnonbreakingstringin thistable	veryverylongnonbreakingstringin thistable	veryverylongnonbreakingstringin thistable

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:growAutofit />
</w:compat>
```

Then the presence of those long non-breaking strings (and the resulting large minimum widths for each table cell) must result in a table width which is then allowed to override the page margins, resulting in the following output:

veryverylongnonbreakingstringinthistable	veryverylongnonbreakingstringinthistable	veryverylongnonbreakingstringi

The resulting table is clipped by the edge of the page on its right side, but the minimum widths of each cell are maintained as defined by the long non-breaking string contents of each. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.23 layoutRawTableWidth (Ignore Space Before Table When Deciding If Table Should Wrap Floating Object)

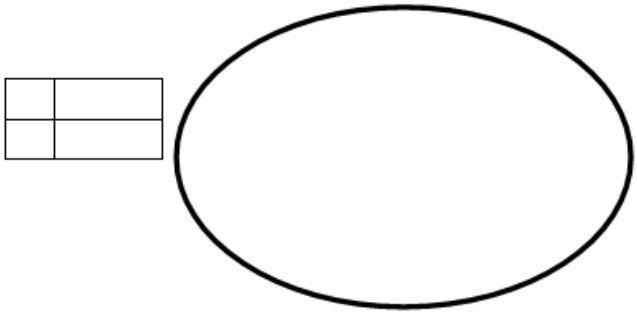
This element specifies how tables which have been indented from the margin using the tblInd element (Part 1, §17.4.51) shall be wrapped around floating objects.

Typically, when a table is positioned next to a floating object, the table shall only remain next to the object if it can fit in the remaining space on the line when considering the full width needed for the table: the space before the table, plus the width of the table. This element, when present with a val attribute value of `true` (or equivalent), specifies that the calculation determining whether the table shall fit next to the object shall not include the space before the table, even if that means that the table is actually clipped by the object.

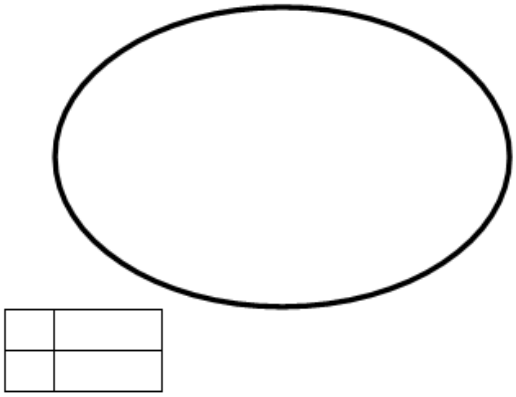
[*Example*: Consider a WordprocessingML document with a floating shape using square wrapping, next to a table which has been indented one inch from the left margin:

```
<w:tbl>
  <w:tblPr>
    <w:tblInd w:w="1440" w:type="dxa" />
  </w:tblPr>
  …
</w:tbl>
```

The resulting presentation would place the table next to the object:



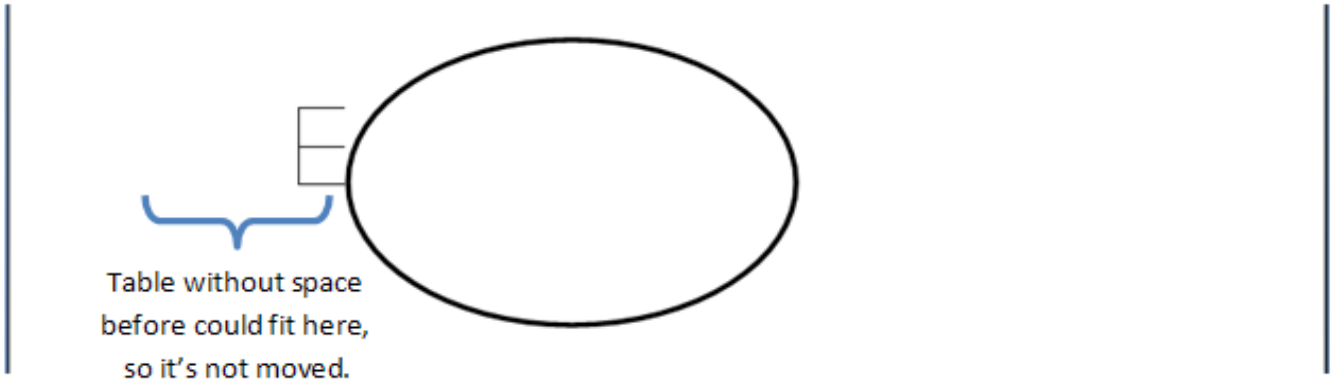
If this object is then moved to the left, such that it would clip the table, the default presentation would have the entire table moved below the shape, since it does not fit in the remaining space on the line:



However, if this compatibility setting is turned on:

```
<w:compat>
  <w:layoutRawTableWidth />
</w:compat>
```

Then the determination to move the table is done ignoring the spaced needed before the table, resulting in the following output:



The resulting table is clipped behind the object, as the fit calculation ignores the space needed before the table. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

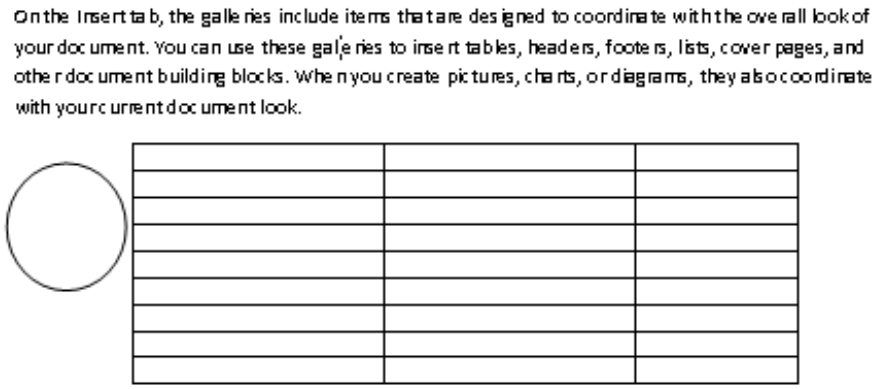
14.7.3.24 layoutTableRowsApart (Allow Table Rows to Wrap Inline Objects Independently)

This element specifies whether tables which are wrapping around floating objects shall wrap around the object as a whole, or if each table row shall individually wrap the object as needed (causing a more stuttered, yet tighter, wrapping of the object).

Typically, when a table wraps around a floating object, the table shall wrap the object as a unit (i.e. the whole table square wraps the object). This element, when present with a `val` attribute value of `true` (or equivalent), specifies that wrapping is applied to each row in the table one by one, even if its means that each row has a different resulting position with respect to the table.

[*Example*: Consider a WordprocessingML document with a floating shape using square wrapping.

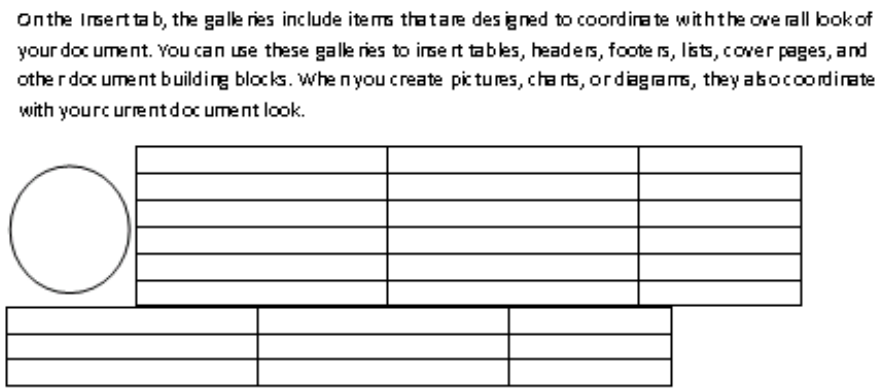
The default presentation would have the entire table wrapping around that shape:



However, if this compatibility setting is turned on:

```
<w:compat>
  <w:layoutTableRowsApart />
</w:compat>
```

Then each row would wrap around the shape one by one, resulting in the following output:



end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

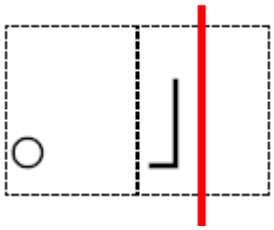
14.7.3.25 lineWrapLikeWord6 (Ignore Compression of Full-Width Punctuation Ending a Line)

This element specifies that applications should ignore the character compression settings specified by the characterSpacingControl element (Part 1, §17.15.1.18) when determining if one more character fits within the text margins on each line of the document. This setting typically results in a character being pushed to the following line, ignoring the fact that the character compression settings would have allowed it to fit within the text boundaries.

Typically, an application would check the character compression settings, and apply any character-level whitespace compression before attempting to fit the last character on the line. This element, when present with a val attribute value of `true` (or equivalent), specifies that applications shall ignore that compression and fit the character as if it should be displayed at its full width, regardless of whether the compression settings are applied.

[*Example*: Consider a paragraph which ends with the following two characters (with each character's bounding box outlined for illustrative purposes:

If the document's character compression settings were not set to `doNotCompress` and text extent fell at the location identified by this red line:



The last character would have compression applied to its blank half, and would fit on the line.

If this compatibility setting is turned on:

```
<w:compat>
  <w:lineWrapLikeWord6 />
</w:compat>
```

Then applications should compress the character, but should treat the character as full width when determining if it fits on the line; in this case, the second character would be displayed on the following line. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.26 mwSmallCaps (Use Specific Small Caps Algorithm)

This element specifies that applications should use a specific algorithm to determine the font size of small caps (the formatting resulting from the use of the smallCaps element (Part 1, §**17.3.2.33**). This emulation typically results in small caps which are smaller than typical small caps at most font sizes.

Typically, applications can utilize any algorithm that results in small caps formatting. This element, when present with a val attribute value of `true` (or equivalent), specifies that applications should determine the font size for small caps using the following algorithm:

- If $font \leq 7$, then the font size for small caps is 7 points.
- Otherwise, sequentially iterate through $sequence$ until $sequence[i] \leq font \leq sequence[i+1]$, at which point the font size for small caps is $sequence[i]$ points.

where

- $sequence$ is an array defined as follows:
  $\{7,9,10,12,14,18,24,36,48,60,72,80, x_1, x_2, \ldots, x_n\}$ where $x_n = 80 + 10 * n$.
- $font$ is an integer calculated as follows:
  The font size of the run to which small caps formatting is applied (in points).

[*Example*: Consider a WordprocessingML document with small caps on its text contents.

If this compatibility setting is turned on:

```
<w:compat>
  <w:mwSmallCaps />
</w:compat>
```

And the font size for a single run is 16 points, and performing the algorithm above would result in 14 points as the calculated font size for small caps. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.27 noColumnBalance (Do Not Balance Text Columns within a Section)

[*Note*: Typically, a continuous section break (Part 1, §17.18.77) balances the content of the previous section, unless the "noColumnBalance" compatibility option is given. *end note*]

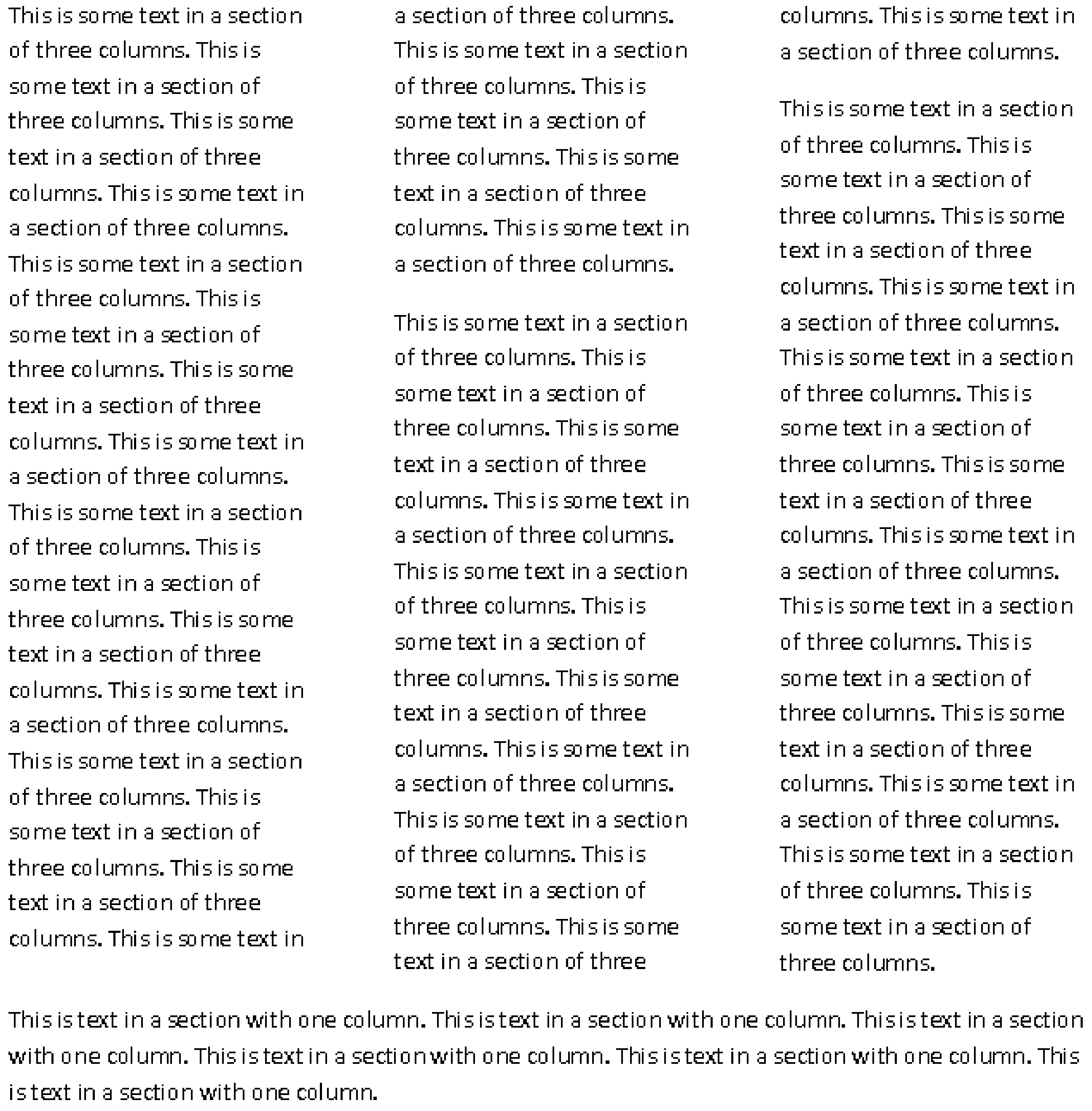
This element specifies whether the contents of sections with multiple columns defined using the cols element (Part 1, §17.6.4) should automatically be balanced. In terms of column layout, *balancing* is the act of attempting to ensure that the number of lines in each column is equivalent (rather than completely filling one column before populating the next).

Typically, column balancing is automatically performed on the contents of sections with multiple columns. This element, when present with a val attribute value of `true` (or equivalent), specifies that column balancing shall not occur, and each column shall be filled individually until the end of the current page, until all text has been displayed, even if this means one or more columns are unused.

[*Example*: Consider a WordprocessingML document with an initial section with three columns, defined by the following section properties:

```
<w:sectPr>
  <w:cols w:num="3" w:space="720" />
</w:sectPr>
```

The default presentation would have the text in that section balanced between those three columns:



This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns.

This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns.

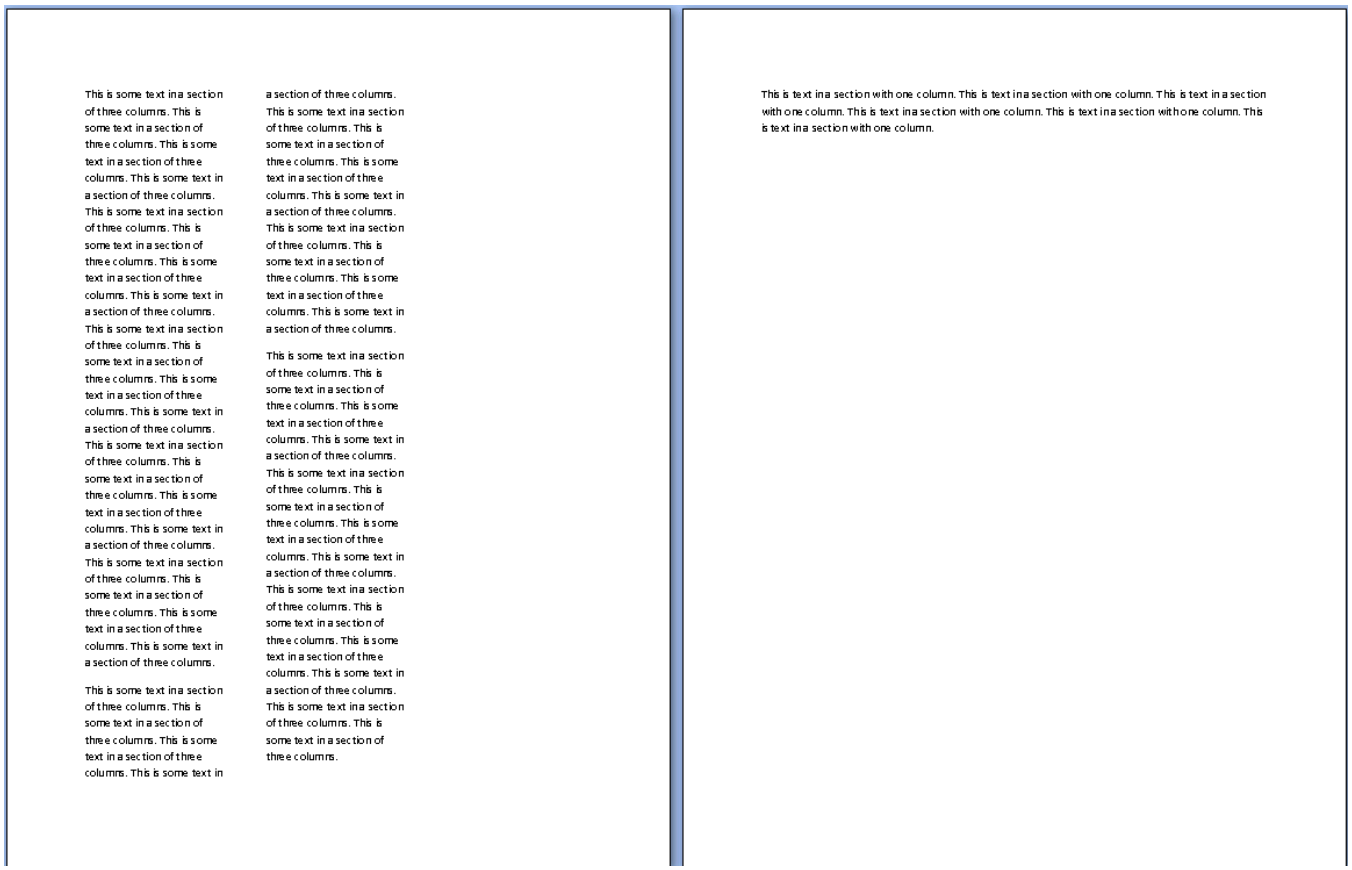
This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns.

This is text in a section with one column. This is text in a section with one column. This is text in a section with one column. This is text in a section with one column. This is text in a section with one column. This is text in a section with one column.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:noColumnBalance />
</w:compat>
```

Then the columns are not balanced, and the contents of the section are used to fill each column to the bottom of the current page in succession, resulting in the following output:



This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns.

This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns.

This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns. This is some text in a section of three columns.

This is text in a section with one column. This is text in a section with one column. This is text in a section with one column. This is text in a section with one column. This is text in a section with one column. This is text in a section with one column.

The next section is now forced to begin on the next page, as the columns on page one extend to the bottom of that page. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.28 noExtraLineSpacing (Do Not Center Content on Lines With Exact Line Height)

This element specifies whether an exact line height using the `spacing` element (Part 1, §17.3.1.33) in the paragraph's properties, each line shall not be automatically centered within the given amount of line spacing.

Typically, if the exact amount of spacing allotted to a line via the paragraph properties exceeds the amount of space required by that line, then the line of text shall be automatically centered when the text of the document is displayed. This element, when present with a `val` attribute value of `true` (or equivalent), specifies thatall additional spacing shall instead be placed below the normal layout of the line of text.

[*Example*: Consider a WordprocessingML document with a line with an exact height of 32 points:

```
<w:p>
  <w:pPr>
    <w:spacing w:line="640" w:lineRule="exact" />
  </w:pPr>
  <w:r>
    <w:t>This is text on a line that's exactly 32 points high.</w:t>
  </w:r>
</w:p>
```

The default presentation would have the resulting text centered on that line:

This is text

This is text on a line that's exactly 32 points high.

This is text.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:noExtraLineSpacing />
</w:compat>
```

Then all line spacing is added after the text, resulting in the following output:

This is text

This is text on a line that's exactly 32 points high.

This is text.

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.29 noLeading (Do Not Add Leading Between Lines of Text)

This element specifies whether the additional leading specified by the current font face shall be added between each line of text when that text is displayed. *Leading* refers to the additional spacing requested by a particular font in order to ensure that letters on subsequent lines do not display in a fashion where they are positioned too closely together.

Typically, leading should be added as specified by the associated font. This element, when present with a val attribute value of `true` (or equivalent), specifies that the additional leading specified by the font shall never be output when the text is displayed.

[*Example*: Consider a WordprocessingML document with three lines of text. The default presentation would have the text displayed as follows:

EXAMPLE TEXT

Some text.
Some text.
Some text.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:noLeading />
</w:compat>
```

Then no leading is added between lines, resulting in the following output:

EXAMPLE TEXT

Some text.
Some text.
Some text.

This adjustment is usually very minute in nature; therefore the result is better illustrated by showing how the characters were pushed out due to the leading added to that text:

EXAMPLE TEXT

Some text.
Some text.
Some text.

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.30 noSpaceRaiseLower (Do Not Increase Line Height for Raised/Lowered Text)

This element specifies whether the height which is allotted to any given line of text when the contents of this document are displayed shall include additional spacing in order to ensure that all raised and/or lowered text can be fully displayed.

Typically, any extra space needed is added to the line to prevent raised and lowered text from being truncated or hidden. This element, when present with a val attribute value of `true` (or equivalent), specifies that the height of the line shall be determined solely by the spacing settings on the parent paragraph, and any raised/lowered text shall just be clipped if it exceeds that space.

[*Example*: Consider a WordprocessingML document with both raised and lowered text. The default presentation would have that text visible:

This is text.

This is text – a lowered word, a raised word.

This is text.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:noSpaceRaiseLower />
</w:compat>
```

Then no additional space should be added to the line height, resulting in the following output:

This is text.

This is text – a lowered word, a raised word.

This is text.

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.31 noTabHangInd (Do Not Create Custom Tab Stop for Hanging Indent)

This element specifies whether applications should always create a hanging indent as a custom tab stop when handling tabs within the contents of a WordprocessingML paragraph. The dontUseIndentAsNumberingTabStop element (§14.7.3.16) specifies if this tab stop shall be used in the case of a tab added as the suffix to numbering in a numbered paragraph, while this element handles the same

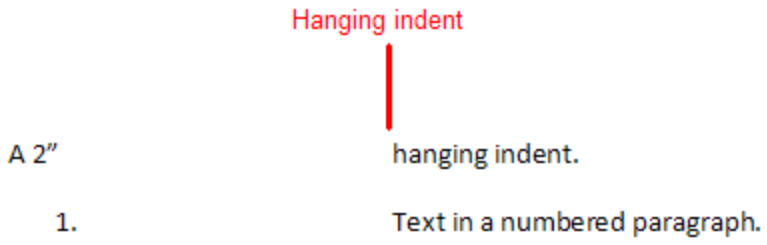
functionality in the generic case (i.e. this element, when set, renders that setting irrelevant as the tab stop is never used).

Typically, the hanging indent on a paragraph shall be treated as a custom tab stop location within that paragraph, allowing the first tab on the first line in the paragraph to advance to the location of the hanging indent. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that no custom tab stop shall be created for a hanging indent on a line under any circumstances.

[*Example*: Consider a WordprocessingML document with two paragraphs (the second numbered, the first not), each with a 2" hanging indent defined as follows (assume the numbering suffix - not shown - is a tab character):

```
<w:p>
  <w:pPr>
    <w:ind w:left="2880" w:hanging="2880" />
  </w:pPr>
  <w:r>
    <w:t>A 2"</w:t>
    <w:tab/>
    <w:t>hanging indent</w:t>
  </w:r>
</w:p>
<w:p>
  <w:pPr>
    <w:numPr>
      <w:ilvl w:val="0" />
      <w:numId w:val="1" />
    </w:numPr>
    <w:ind w:left="2880" w:hanging="2880" />
  </w:pPr>
  <w:r>
    <w:t>Text in a numbered paragraph.</w:t>
  </w:r>
</w:p>
```

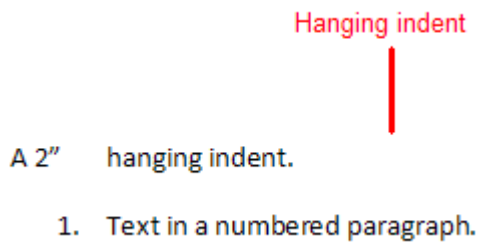
The default presentation would have both the numbering and the tab in the regular paragraph advancing to the 2" custom tab stop generated by the hanging indent:



However, if this compatibility setting is turned on:

```
<w:compat>
  <w:noTabHangInd />
</w:compat>
```

Then no tab stop exists at 2", and therefore the tab stops must advance to the location of the next automatic tab stop for this document (which is set to occur every 0.5"), resulting in the following output:



end example]

This element’s content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.32 printBodyTextBeforeHeader (Print Body Text before Header/Footer Contents)

This element specifies the order in which the contents of the main document story and any headers and/or footers shall be sent to the printer.

Typically, the contents of a document are sent to the printer as follows:

- First, the contents of headers/footers are sent to the printer
- Finally, the contents of the main document story are sent to the printer

This element, when present with a `val` attribute value of `true` (or equivalent), specifies that this order shall be reversed, and that the body text shall be sent to the printer before any header/footer text. This reversal allows for the processing of PostScript codes in the text layer in the same order as afforded by some legacy word processing applications.

[*Example*: Consider a WordprocessingML document which is printed. The default resulting print order is the headers and footers for each page, followed by the page contents.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:printBodyTextBeforeHeader />
</w:compat>
```

Then this order must be reversed, and the page contents must be printed before the corresponding header and/or footer for each page. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.33 printColBlack (Print Colors as Black And White without Dithering)

This element specifies the way in which colored text and/or objects shall be handled when printed to a printer whose printer settings indicate that it can only handle black and white text.

Typically, the contents of a colored document are sent to a black and white printer using grayscale (different shades of gray) to represent each of the possible colors. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that colors are not printed as mapped shades of grey, but rather exclusively in solid black and white. This setting prevents the fuzzy look that can occur when gray or blue content is dithered. *Dithering* is the process by which colors are simulated using various patterns of black dots on a white background

[*Example*: Consider a WordprocessingML document which is printed to a black and white printer. The default resulting printed content is typically dithered to appear in the appropriate shade of grayscale text.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:printColBlack />
</w:compat>
```

Then the page contents must be printed as exclusively black or exclusively white text as needed, and no grayscale output must occur. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.34 selectFldWithFirstOrLastChar (Select Field When First or Last Character Is Selected)

This element specifies whether applications should automatically select the entire contents of a field in a WordprocessingML document when the first or last character is selected.

Typically, users can select any character individually within the result of a field in the document. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that selecting the first or last character of that field result shall automatically result in the selection of the entire field.

[*Example*: Consider a WordprocessingML document which contains the following (with a field marked in gray shading):

Author Tristan Davis would like to welcome you.

The default presentation would allow the first character of that field to be selected:

Author Tristan Davis would like to welcome you.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:selectFldWithFirstOrLastChar />
</w:compat>
```

Then that selection would automatically result in the entire field being selected, resulting in the following:

Author Tristan Davis would like to welcome you.

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.35 shapeLayoutLikeWW8 (Ignore Text Wrapping around Objects at Bottom of Page)

This element specifies that applications should ignore the line wrapping setting specified by a floating object, instead allowing text to be displayed beneath it under the specific set of conditions identified below.

Typically, text wrapping around a floating object is dictated by the presence of one of the following as a child element of the object's anchor element (Part 1, §20.4.2.3):

- wrapNone (Part 1, §20.4.2.15) element, which specifies no text wrapping
- wrapSquare (Part 1, §20.4.2.17) element, which specifies square text wrapping
- wrapThrough (Part 1, §20.4.2.18) element, which specifies through text wrapping
- wrapTight (Part 1, §20.4.2.19) element, which specifies tight text wrapping
- wrapTopAndBottom (Part 1, §20.4.2.19) element, which specifies top and bottom text wrapping

This element, when present with a val attribute value of `true` (or equivalent), specifies that applications shall allow text to wrap beneath a floating object, ignoring the object's true wrapping setting, when the following conditions are met:

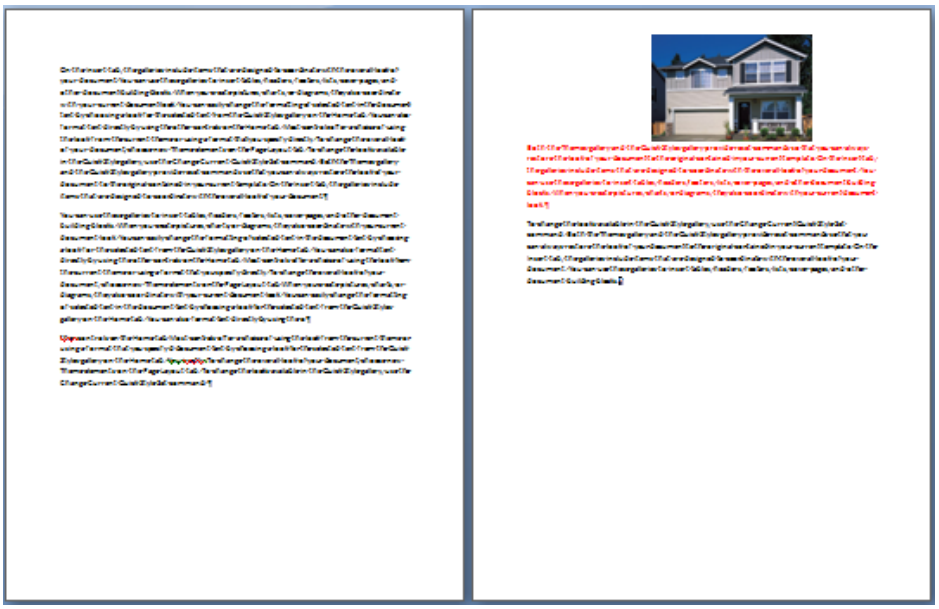
- The floating object has any of the following elements present as a child of the object's anchor element: wrapSquare, wrapTight, or wrapTopAndBottom.
- The floating object has a positionV element (Part 1, §20.4.2.11) with a relativeFrom attribute value of `line`.
- The floating object has a negative value for the child posOffset element (Part 1, §20.4.2.12) of the positionV element.
- The paragraph containing the anchor element would appear directly after the previous paragraph if the wrapping settings were ignored.

- The paragraph containing the anchor element would be pushed to the next page if the wrapping settings were respected.

[*Example*: Consider a WordprocessingML document containing a DrawingML object which meets the conditions outlined above:

```
<w:p>
  <w:r>
    <w:t>Sample text. Sample text. Sample text. Sample text. Sample text. Sample
text.</w:t>
  </w:r>
  <w:r>
    <w:drawing>
      <wp:anchor … >
        <wp:positionV relativeFrom="line">
          <wp:posOffset>-428914</wp:posOffset>
        </wp:positionV>
        <wp:wrapTopAndBottom />
        …
      </wp:anchor>
    </w:drawing>
  </w:r>
  <w:r>
    <w:t> Sample text. Sample text. Sample text. Sample text. Sample text.
Sample text.</w:t>
  </w:r>
  …
</w:p>
```

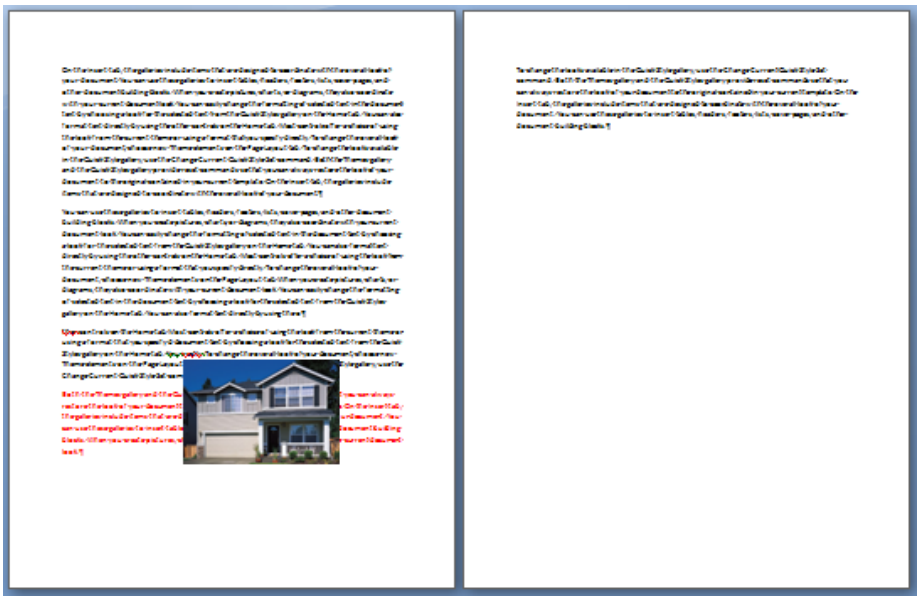
When the wrapping settings are respected, the shape and its paragraph do not fit on the page, so they are moved to the next page (the paragraph containing the anchor has been highlighted for illustrative purposes):



If this compatibility setting is turned on:

```
<w:compat>
  <w:shapeLayoutLikeWW8 />
</w:compat>
```

Then applications should ignore the wrapping setting and allow text to wrap below the object. This behaviour results in the following (again, the paragraph containing the anchor has been highlighted for illustrative purposes):



end example]

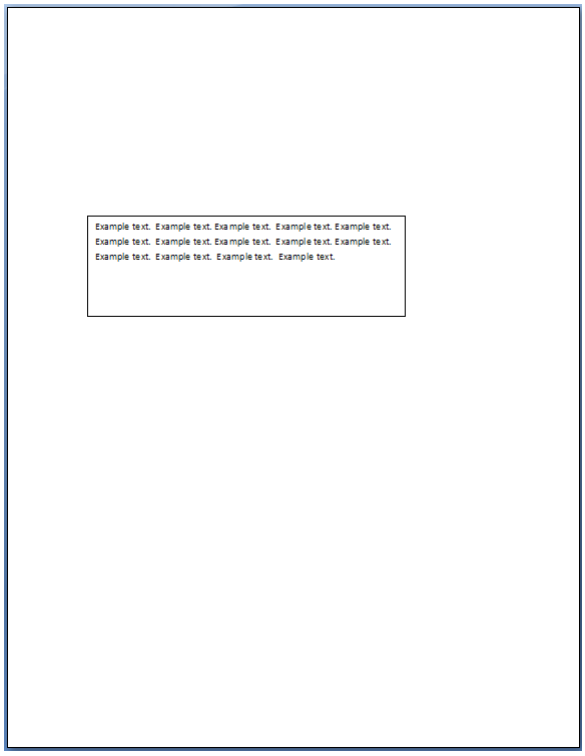
This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.36 showBreaksInFrames (Display Page/Column Breaks Present in Frames)

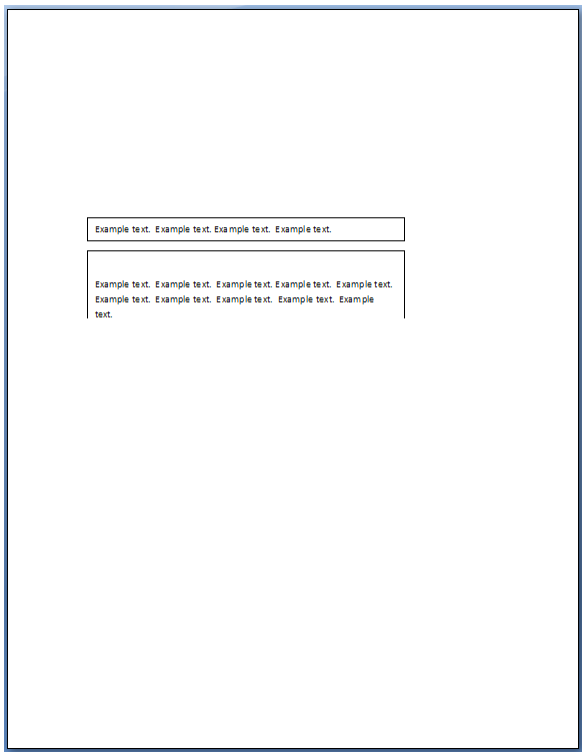
This element specifies whether applications should honor the presence of page and/or column breaks which are present within the contents of paragraphs which have been defined as frames using the `framePr` element (Part 1, §17.3.1.11).

Typically, breaks within frames shall be ignored and shall have no effect on the display of the paragraph in which they are contained. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that rather than completely ignoring these breaks, applications should display the break and move the remaining frame content, and all subsequent text, to the next page and/or column, as needed.

[*Example*: Consider a WordprocessingML document with a paragraph contained within a text frame:



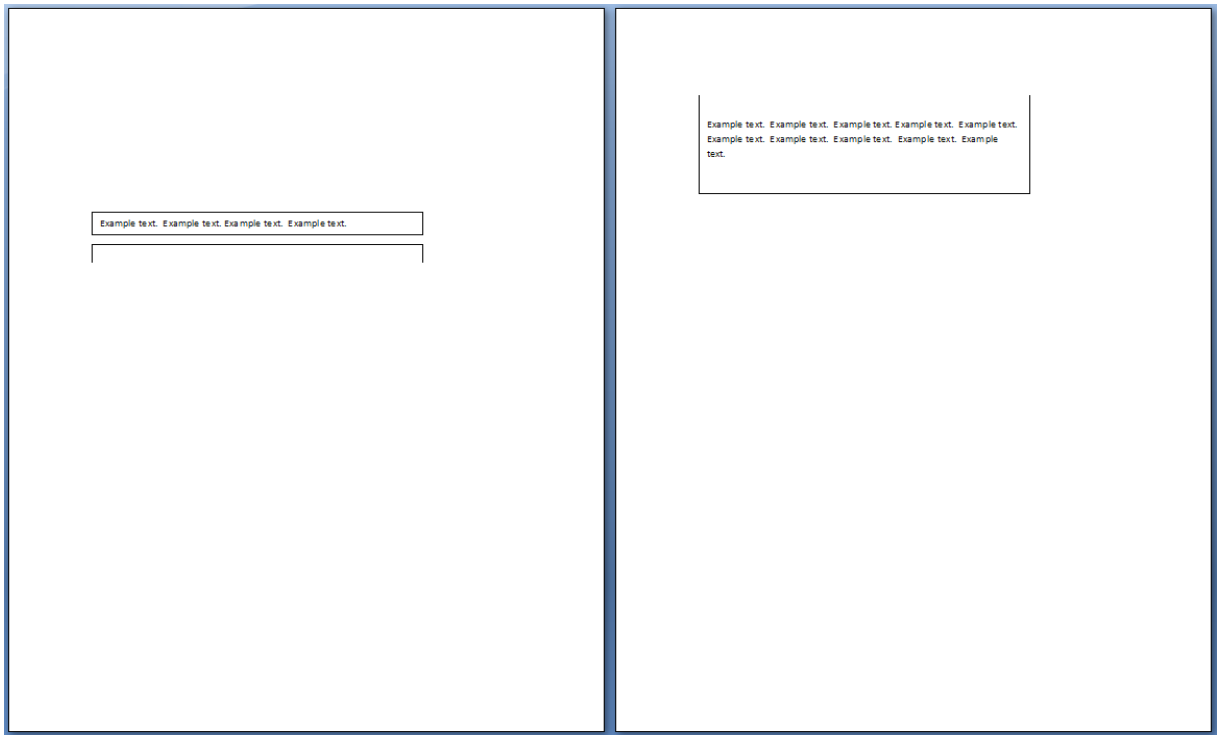
The default presentation would display the page break inline in the frame (breaking the frame into two) but would not actually break the page:



However, if this compatibility setting is turned on:

```
<w:compat>
  <w:showBreaksInFrames />
</w:compat>
```

Then the page breaks is used even though they are present in the frame, breaking the page and resulting in the following output:



end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.37 spacingInWholePoints (Only Expand/Condense Text By Whole Points)

This element specifies how applications should apply text expansion/compression defined using the spacing element (Part 1, §17.3.2.35) within a set of run properties.

Typically, as defined in the spacing element, text within runs in a WordprocessingML document can be expanded or compressed in increments of twentieths of a point. This element, when present with a val attribute value of `true` (or equivalent), specifies that the expansion and compression of text shall only be performed in increments of points. Any value which is not equal to an expansion or compression of a whole point shall be rounded down to the nearest whole point when the text is expanded/compressed within the WordprocessingML document.

[*Example*: Consider a WordprocessingML document with three paragraphs of text, each expanded by a varying amount, as follows:

```
<w:p>
  …
  <w:r>
    <w:t>This is text.</w:t>
  </w:r>
</w:p>
```



```
<w:p>
  …
  <w:r>
    <w:rPr>
      <w:spacing w:val="20" />
    </w:rPr>
    <w:t>This is text.</w:t>
  </w:r>
</w:p>
<w:p>
  …
  <w:r>
    <w:rPr>
      <w:spacing w:val="36" />
    </w:rPr>
    <w:t>This is text.</w:t>
  </w:r>
</w:p>
```

The default presentation would have each run of text expanded exactly as requested:

Regular Text:	This is text.
Text expanded by 1 point:	This is text.
Text expanded by 1.8 points:	This is text.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:spacingInWholePoints />
</w:compat>
```

Then the third line - with an expansion of 1.8 points - would instead be rounded down to the nearest whole number of points when expanded, resulting in the following output:

Regular Text:	This is text.
Text expanded by 1 point:	This is text.
Text expanded by 1.8 points:	This is text.

In the resulting output, the second and third lines are identical, as the third line has a next expansion of exactly one point. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.38 splitPgBreakAndParaMark (Always Move Paragraph Mark to Page after a Page Break)

This element specifies whether a page break shall automatically complete the line on which it appears, moving the end of the paragraph to a new line on the next page, or if it shall behave as true run-level content within its current paragraph.

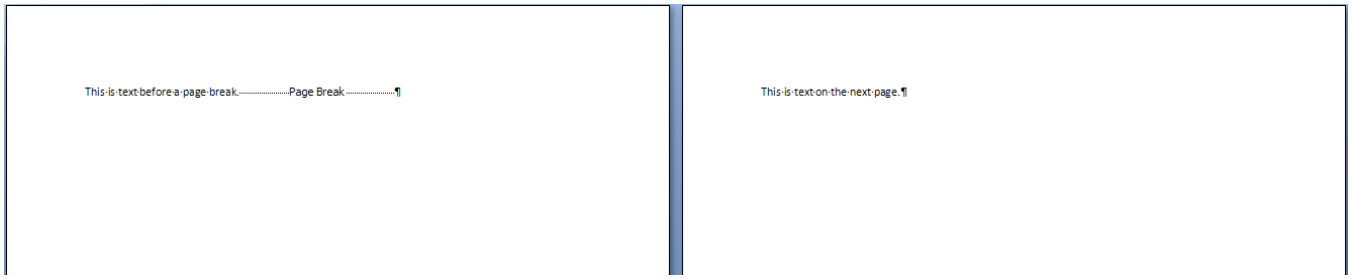
Typically, a page break defined using the `br` element (Part 1, §17.3.3.1) is treated as run-level content, which means that although it delimits the end of the page, if there is no content after it within the current paragraph, that the paragraph shall also end on that page. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that a page break shall always immediately end the current page, moving the paragraph mark which delimits the end of its parent paragraph to a new line on the next page.

Note that this setting only affects the case where there is no run-level content after the page break within the paragraph - if any further run content appears in the paragraph it shall appear on subsequent lines on the next page.

[*Example*: Consider a WordprocessingML document with two paragraphs of content - the first ending with a page break:

```
<w:p>
  <w:r>
    <w:t>This is text before a page break.</w:t>
    <w:br w:type="page" />
  </w:r>
</w:p>
<w:p>
  <w:r>
    <w:t>This is text on the next page.</w:t>
  </w:r>
</w:p>
```

The default presentation would have the text content `This is text on the next page.` as the first line of the second page, as there is no run content after the page break in paragraph one, and therefore no need for a new line on page two (in this image, a graphical illustration of the pilcrow and the page break have been added for clarity):



However, if this compatibility setting is turned on:

```
<w:compat>
  <w:splitPgBreakAndParaMark />
</w:compat>
```

Then even though it is followed by no additional content, the page break must immediately end the first page, pushing the end of the first paragraph onto the first line of the second page, resulting in the following output:

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.39 subFontBySize (Require Exact Size During Font Substitution)

This element specifies whether applications shall accept fonts which cannot be rendered at the size specified by the sz (Part 1, §17.3.2.38) and/or szCs (Part 1, §17.3.2.39) elements on the parent run when performing font substitution.

Typically, applications can perform font substitution as defined in Part 1, §17.8.2, with no additional restrictions. This element, when present with a val attribute value of `true` (or equivalent), specifies that when a potential substitute font has been located, an application shall check whether that font is capable of displaying characters at the specified point size. If it is not, that font is not considered as a substitute font (i.e. it is rejected, and the next closest match is considered).

[*Example*: Consider a WordprocessingML document with a series of characters in an unavailable font. The default presentation would use any method used by the application to perform that font substitution.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:subFontBySize />
</w:compat>
```


For each run, the application determines if the substitute font produced by its font substitution algorithm can be displayed at the size specified by the run's sz and/or szCs elements. If it cannot, that font is not used and the next closest match as substitute font is considered. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.40 suppressBottomSpacing (Ignore Exact Line Height for Last Line on Page)

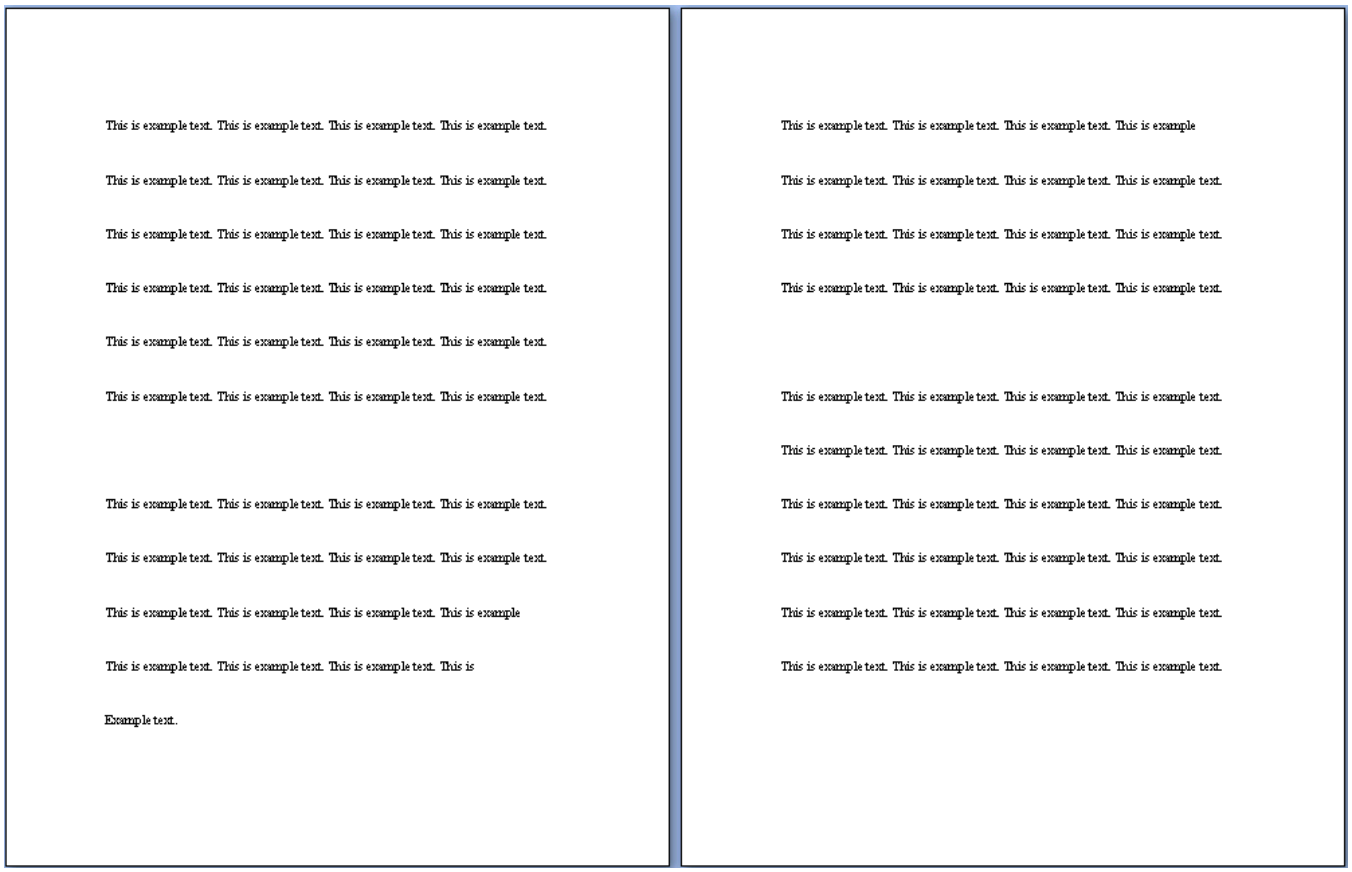
This element specifies whether an exact line height specified using the spacing element (Part 1, §17.3.1.33) with a lineRule attribute value of `exact` shall be ignored for the last line on each page.

Typically, if an exact line height has been specified using the spacing element, then all lines within that paragraph have the necessary line spacing added to them in order to meet this constraint. This element, when present with a val attribute value of `true` (or equivalent), specifies that no additional spacing shall be added below the last line on each page as a result of these line spacing requirements - a line shall be placed on the bottom of the page if its characters fit on that page ignoring the necessary space after.

[*Example*: Consider a WordprocessingML document whose first paragraph has a line spacing setting requiring exactly 48 points of space per line:

```
<w:p>
  <w:pPr>
    <w:spacing w:line="960" w:lineRule="exact" />
  </w:pPr>
  …
</w:p>
```

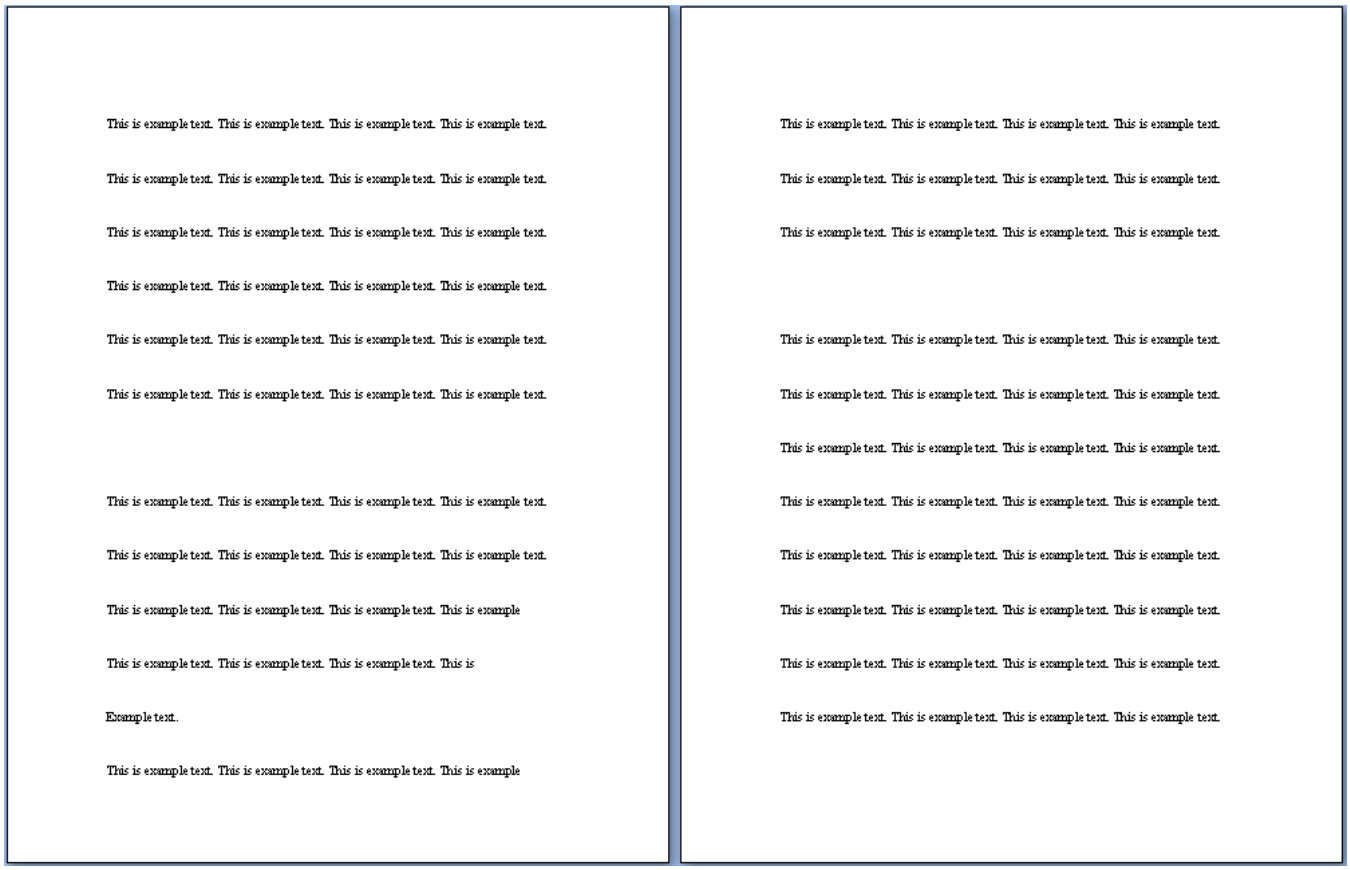
The default presentation would have the necessary amount of space added between each line such that all lines in the paragraph are centered within 48 points of spacing:



However, if this compatibility setting is turned on:

```
<w:compat>
  <w:suppressBottomSpacing />
</w:compat>
```

Then that constraint must be lifted for the last line on the page (although all other lines are unaffected), resulting in the following output:



The first line from the following page was moved on the first page, as without being subjected to the line height constraint, it is possible to fit it at the bottom of the first page. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.41 suppressSpacingAtTopOfPage (Ignore Minimum Line Height for First Line on Page)

This element specifies whether the minimum line height specified using the spacing element (Part 1, §17.3.1.33) with a lineRule attribute value of `atLeast` shall be ignored for the first line on each page.

Typically, if a minimum line height has been specified using the spacing element, then all lines within that paragraph have the necessary line spacing added to them in order to meet this constraint. This element, when present with a val attribute value of `true` (or equivalent), specifies that no additional spacing shall be added above the first line on each page as a result of this line spacing requirements - the top of the text characters on the first line shall be at the top edge of the page.

[*Example*: Consider a WordprocessingML document whose first paragraph has a line spacing setting requiring at least 25 points of space per line:


```
<w:p>
  <w:pPr>
    <w:spacing w:line="500" w:lineRule="atLeast" />
  </w:pPr>
  …
</w:p>
```

The default presentation would have the necessary amount of space added between each line such that all lines in the paragraph are centered within 25 points of spacing (highlighting has been added to the image below in order to illustrate the additional spacing above the first line):

Example text. Example text. Example text. Example text. Example text. Example text. Example text.

Example text. Example text. Example text. Example text. Example text. Example text. Example text.

Example text. Example text. Example text. Example text. Example text. Example text. Example text.

Example text. Example text. Example text. Example text. Example text. Example text. Example text.

Example text. Example text. Example text. Example text.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:suppressSpacingAtTopOfPage />
</w:compat>
```

Then no additional line spacing must be added above the first line on the page (although all other lines are unaffected), resulting in the following output:

Example text. Example text. Example text. Example text. Example text. Example text. Example text.

Example text. Example text. Example text. Example text. Example text. Example text. Example text.

Example text. Example text. Example text. Example text. Example text. Example text. Example text.

Example text. Example text. Example text. Example text. Example text. Example text. Example text.

Example text. Example text. Example text. Example text.

However, if this line spacing constraint was exactly 25 points, then this setting would have no effect:

Example text. Example text. Example text. Example text. Example text. Example text. Example text.

Example text. Example text. Example text. Example text. Example text. Example text. Example text.

Example text. Example text. Example text. Example text. Example text. Example text. Example text.

Example text. Example text. Example text. Example text. Example text. Example text. Example text.

Example text. Example text. Example text. Example text.

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.42 suppressSpBfAfterPgBrk (Do Not Use Space Before On First Line After a Page Break)

This element specifies that applications should not postpone any before paragraph spacing to the first line containing content after a page break.

Typically, a page break defined using the `br` element (Part 1, §17.3.3.1) is treated as run-level content, which means that although it delimits the end of the page, if there is no content after it within the current paragraph, that the paragraph shall also end on that page. However, in the case where there is additional run-level content within the same paragraph, that content, although part of the same paragraph as the page break, is displayed on the following page.

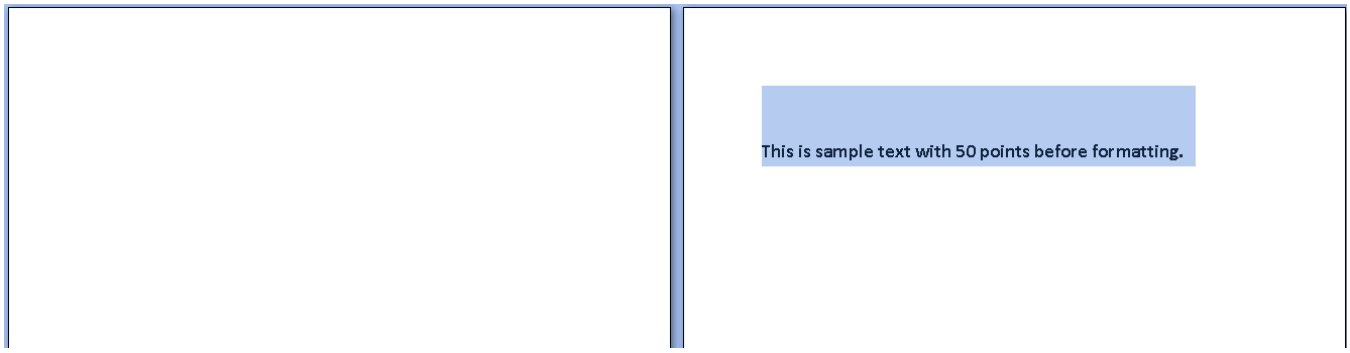
This leads to a situation where the only run content on the page with the page break is the break itself, with all subsequent content on the following page. In this case, applications shall apply the value specified by the `spacing` element's `before` attribute to the first line on the new page (since it is ostensibly the only page with content in that paragraph).

This element, when present with a `val` attribute value of `true` (or equivalent), specifies the paragraph before spacing shall not be 'postponed' in this way - if the line with the page break has no content, then the `spacing` element's `before` attribute is simply ignored.

[*Example*: Consider a WordprocessingML document whose first paragraph specifies that it must be preceded by 50 points of additional spacing:


```
<w:p>
  <w:pPr>
    <w:spacing w:before="1000" />
  </w:pPr>
  <w:r>
    <w:br w:type="page" />
    <w:t>This is sample text with 50 points before formatting.</w:t>
  </w:r>
</w:p>
```

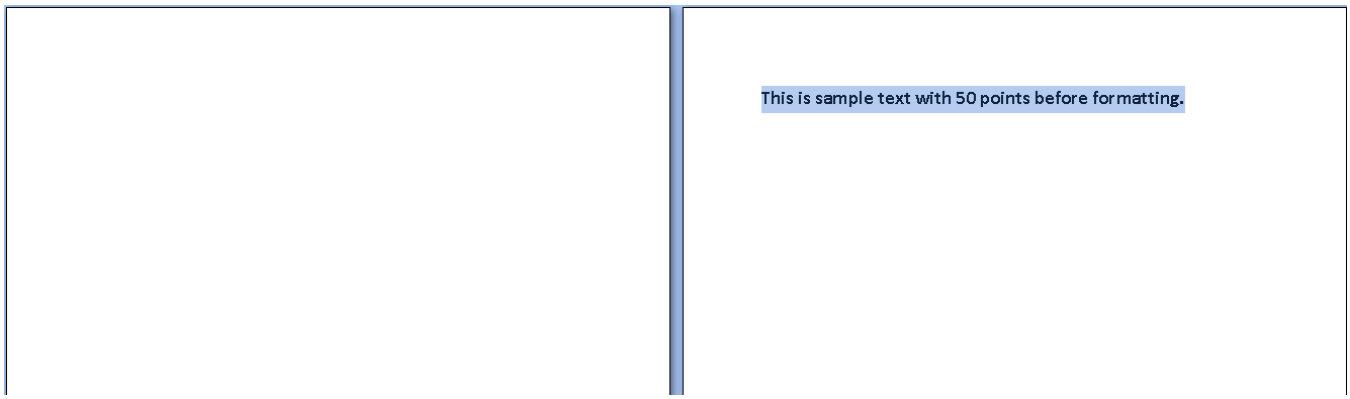
The default presentation would have the necessary amount of space added to the first line on the second page, as the page break was not preceded by any run content (highlighting has been added to the image below in order to illustrate the additional spacing above the first line):



However, if this compatibility setting is turned on:

```
<w:compat>
  <w:suppressSpBfAfterPgBrk />
</w:compat>
```

Then the spacing must not be added above the first line on the page (it is essentially ignored), resulting in the following output:



end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.43 suppressTopSpacing (Ignore Minimum and Exact Line Height for First Line on Page)

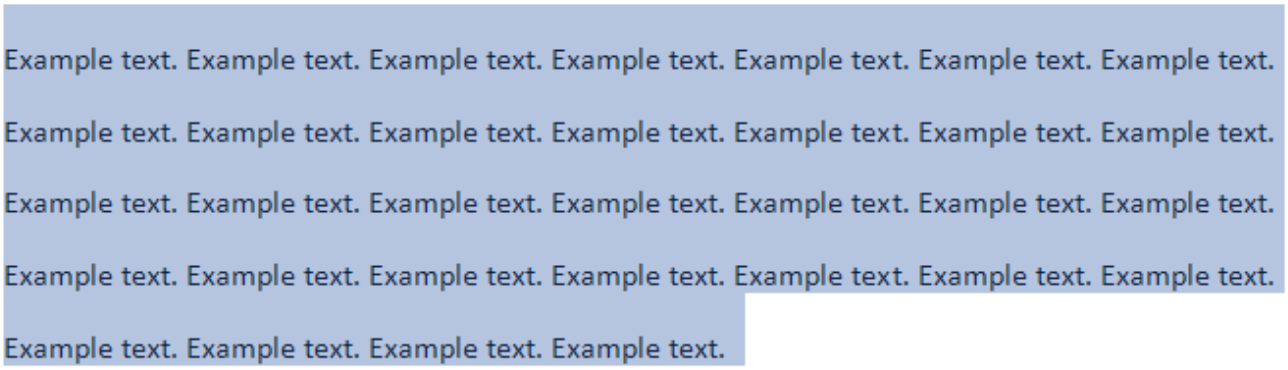
This element specifies whether the minimum line height specified using the spacing element (Part 1, §17.3.1.33) with a lineRule attribute value of `atLeast` or `exact` shall be ignored for the first line on each page.

Typically, if a minimum or exact line height has been specified using the spacing element, then all lines within that paragraph have the necessary line spacing added to them in order to meet this constraint. This element, when present with a val attribute value of `true` (or equivalent), specifies that no additional spacing shall be added above the first line on each page as a result of these line spacing requirements - the top of the text characters on the first line shall be at the top edge of the page.

[*Example*: Consider a WordprocessingML document whose first paragraph has a line spacing setting requiring exactly 25 points of space per line:

```
<w:p>
  <w:pPr>
    <w:spacing w:line="500" w:lineRule="exact" />
  </w:pPr>
  …
</w:p>
```

The default presentation would have the necessary amount of space added between each line such that all lines in the paragraph are centered within 25 points of spacing (highlighting has been added to the image below in order to illustrate the additional spacing above the first line):



However, if this compatibility setting is turned on:

```
<w:compat>
  <w:suppressTopSpacing />
</w:compat>
```


Then no additional line spacing must be added above the first line on the page (although all other lines are unaffected), resulting in the following output:

Example text. Example text. Example text. Example text. Example text. Example text. Example text.

Example text. Example text. Example text. Example text. Example text. Example text. Example text.

Example text. Example text. Example text. Example text. Example text. Example text. Example text.

Example text. Example text. Example text. Example text. Example text. Example text. Example text.

Example text. Example text. Example text. Example text.

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.44 suppressTopSpacingWP (Use Static Text Leading)

(The terms *baseline to baseline distance* and *unitsPerEm*, used below, are defined in ISO/IEC 14496-22.)

This element specifies that applications should use the values defined below to calculate the baseline to baseline distance (BTBD) in this document. This can result in lines appearing slightly condensed vertically.

Without this setting, applications calculate baseline to baseline distance using the metrics defined by ISO/IEC 14496-22. This element, when present with a val attribute value of `true` (or equivalent), specifies that applications should calculate this as follows:

$$BTBD = \text{unitsPerEm} + 2\text{pt}$$

[*Example*: If this compatibility setting is turned on:

```
<w:compat>
  <w:suppressTopSpacingWP />
</w:compat>
```

Then applications use a baseline to baseline distance as calculated before. With a 16 point font, this would result in a baseline to baseline distance of 18 points. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.45 swapBordersFacingPages (Swap Paragraph Borders on Odd Numbered Pages)

This element specifies whether left and right paragraph borders defined under the pBdr element (Part 1, §17.3.1.24) shall be swapped under conditions where it is possible that the those pages are intended to be used to create a book-like publication.

Typically, no changes shall be made to the positions of paragraph borders defined under the pBdr element - a right border is always on the right, and a left border is always on the left. This element, when present with a val attribute value of true (or equivalent), specifies that under the two following conditions:

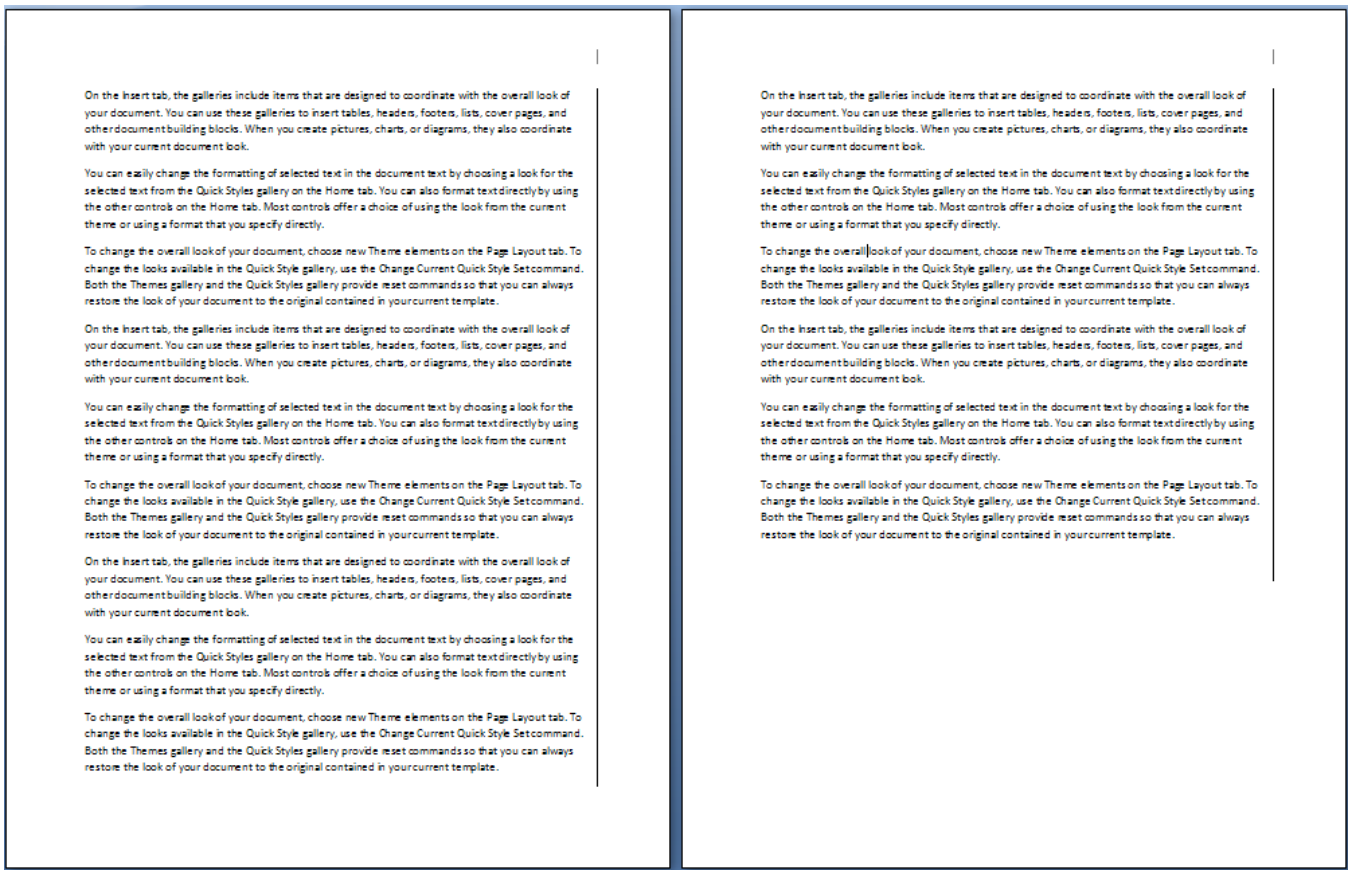
- The margins in this document are mirrored using the mirrorMargins element (Part 1, §17.15.1.57)
- The header/footers in this document are different on even and odd numbered pages using the evenAndOddHeaders element (Part 1, §17.10.1)

That paragraph borders on odd-numbered pages are swapped - that is, left borders shall be displayed on the right and right borders shall be displayed on the left.

[*Example*: Consider a WordprocessingML document for which the mirrorMargins element is present, and whose default paragraph style includes a paragraph border to be displayed on the right side of each paragraph:

```
<w:style w:type="paragraph" w:default="1" w:styleId="Normal" >
  …
  <w:pPr>
    <w:pBdr>
      <w:right w:val="single" w:color="auto" />
    </w:pBdr>
    …
  </w:pPr>
</w:style>
```

If a two-page document is created using this default paragraph style, then all paragraphs has a border on the right side, as follows:



On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks. When you create pictures, charts, or diagrams, they also coordinate with your current document look.

You can easily change the formatting of selected text in the document text by choosing a look for the selected text from the Quick Styles gallery on the Home tab. You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly.

To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command. Both the Themes gallery and the Quick Styles gallery provide reset commands so that you can always restore the look of your document to the original contained in your current template.

On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks. When you create pictures, charts, or diagrams, they also coordinate with your current document look.

You can easily change the formatting of selected text in the document text by choosing a look for the selected text from the Quick Styles gallery on the Home tab. You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly.

To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command. Both the Themes gallery and the Quick Styles gallery provide reset commands so that you can always restore the look of your document to the original contained in your current template.

On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks. When you create pictures, charts, or diagrams, they also coordinate with your current document look.

You can easily change the formatting of selected text in the document text by choosing a look for the selected text from the Quick Styles gallery on the Home tab. You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly.

To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command. Both the Themes gallery and the Quick Styles gallery provide reset commands so that you can always restore the look of your document to the original contained in your current template.

On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks. When you create pictures, charts, or diagrams, they also coordinate with your current document look.

You can easily change the formatting of selected text in the document text by choosing a look for the selected text from the Quick Styles gallery on the Home tab. You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly.

To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command. Both the Themes gallery and the Quick Styles gallery provide reset commands so that you can always restore the look of your document to the original contained in your current template.

On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks. When you create pictures, charts, or diagrams, they also coordinate with your current document look.

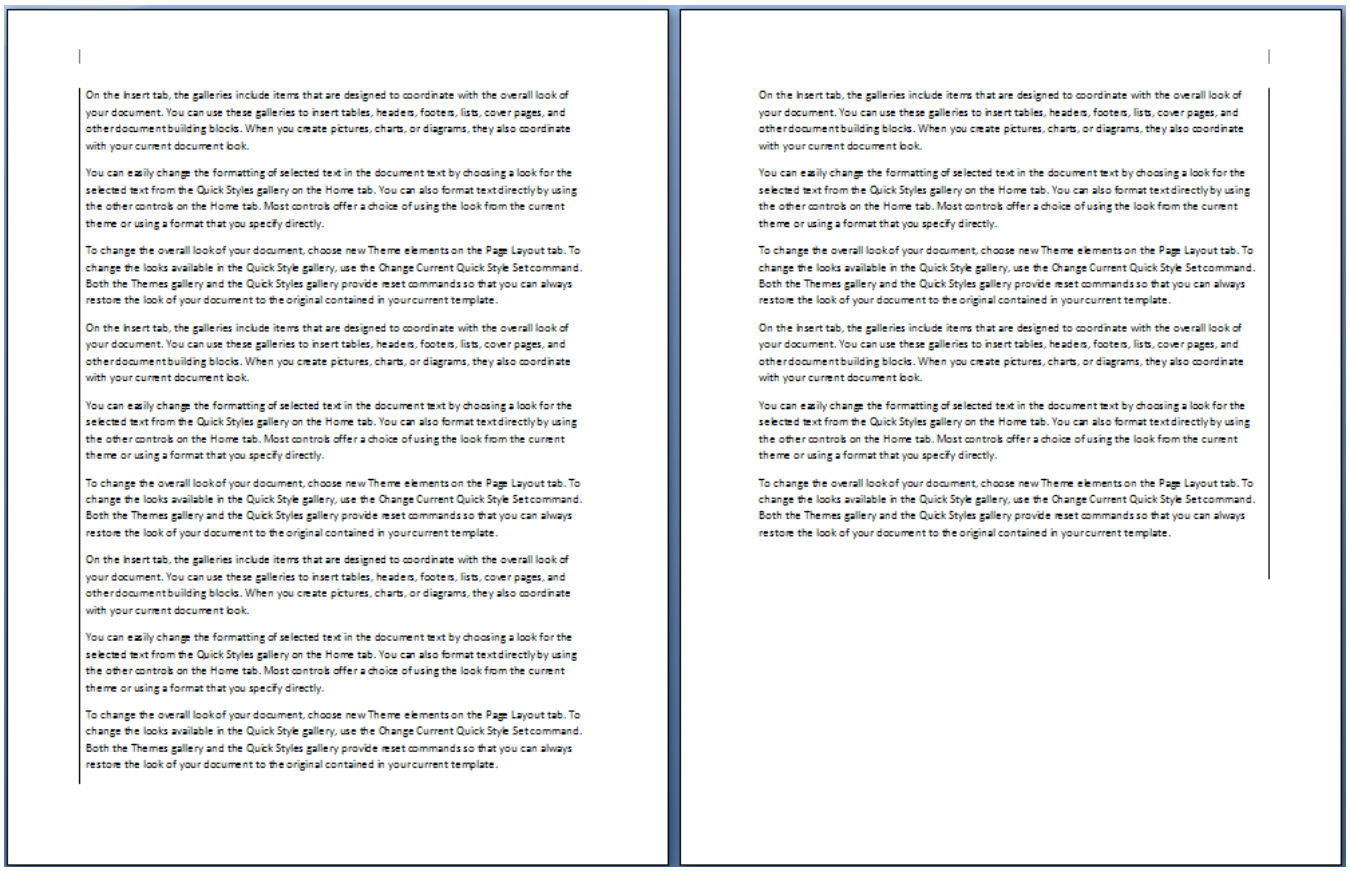
You can easily change the formatting of selected text in the document text by choosing a look for the selected text from the Quick Styles gallery on the Home tab. You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly.

To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command. Both the Themes gallery and the Quick Styles gallery provide reset commands so that you can always restore the look of your document to the original contained in your current template.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:swapBordersFacingPages />
</w:compat>
```

Then the borders on the first page (being an odd-numbered page) must be swapped, resulting in the following output:



On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks. When you create pictures, charts, or diagrams, they also coordinate with your current document look.

You can easily change the formatting of selected text in the document text by choosing a look for the selected text from the Quick Styles gallery on the Home tab. You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly.

To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command. Both the Themes gallery and the Quick Styles gallery provide reset commands so that you can always restore the look of your document to the original contained in your current template.

On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks. When you create pictures, charts, or diagrams, they also coordinate with your current document look.

You can easily change the formatting of selected text in the document text by choosing a look for the selected text from the Quick Styles gallery on the Home tab. You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly.

To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command. Both the Themes gallery and the Quick Styles gallery provide reset commands so that you can always restore the look of your document to the original contained in your current template.

On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks. When you create pictures, charts, or diagrams, they also coordinate with your current document look.

You can easily change the formatting of selected text in the document text by choosing a look for the selected text from the Quick Styles gallery on the Home tab. You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly.

To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command. Both the Themes gallery and the Quick Styles gallery provide reset commands so that you can always restore the look of your document to the original contained in your current template.

On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks. When you create pictures, charts, or diagrams, they also coordinate with your current document look.

You can easily change the formatting of selected text in the document text by choosing a look for the selected text from the Quick Styles gallery on the Home tab. You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly.

To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command. Both the Themes gallery and the Quick Styles gallery provide reset commands so that you can always restore the look of your document to the original contained in your current template.

On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks. When you create pictures, charts, or diagrams, they also coordinate with your current document look.

You can easily change the formatting of selected text in the document text by choosing a look for the selected text from the Quick Styles gallery on the Home tab. You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly.

To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command. Both the Themes gallery and the Quick Styles gallery provide reset commands so that you can always restore the look of your document to the original contained in your current template.

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.46 truncateFontHeightsLikeWP6 (Use Truncated Integer Division For Font Calculation)

This element specifies that applications should perform a specific method of calculation when converting font heights, specified in points using the sz (Part 1, §17.3.2.38) and szCs (Part 1, §17.3.2.39) elements, into pixels. This algorithm often results in a smaller than typical visual appearance of text for a given point size.

Typically, applications convert points to pixels using any approximate mathematical conversion mechanism (often, rounded integer division). This element, when present with a val attribute value of `true` (or equivalent), specifies that applications should use truncated integer division when performing this conversion (any non-integer value is truncated to determine the integer value resulting from the conversions).

[*Example*: If this compatibility setting is turned on:

```
<w:compat>
  <w:truncateFontHeightsLikeWP6 />
</w:compat>
```

Then applications shall use truncated integer division when calculating the height of characters.

For example, if the conversion is done as follows:

$$sz_{px} = sz_{pt} * N \frac{px}{inch} * \frac{1\ inch}{72\ pt}$$

where:

- sz_{pt} = size in points
- sz_{px} = size in pixels
- N = resolution in pixels per inch

Converting a 14 point font on a 96 dpi device results in $sz_{px} = 14 * 96 * \frac{1}{72} = 18\frac{2}{3}px$. If this setting is on, the result is truncated and the font is displayed using 18 pixels, even though 19 would be closer to the actual value. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.47 underlineTabInNumList (Underline Following Character Following Numbering)

This element specifies whether applications shall underline the character following the numbering defined using the suff element (Part 1, §17.9.29) when both the numbering itself and the first letter of the corresponding numbered paragraph is underlined.

Typically, the tab or space character generated between numbering and the corresponding paragraph of text is never formatted, since it is automatically generated by the suff element. This element, when present with a val attribute value of `true` (or equivalent), specifies that the tab or space shall tab or space shall be underlined the same way as the numbering symbol itself in the following conditions:

- The numbering is underlined
- The first character of the paragraph is underlined

[*Example*: Consider a WordprocessingML document with two numbered paragraphs: one with underlined text and the other without. The default presentation would have the tab characters free of underlining in both cases:

1. Example Text

2. Example Text

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:underlineTabInNumList />
</w:compat>
```

Then the second paragraph meets the criteria defined above for having the suffix character underlined, resulting in the following output:

1. Example Text
2. Example Text

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.48 useAltKinsokuLineBreakRules (Use Alternate Set of East Asian Line Breaking Rules)

This element specifies an alternate set of characters which can be used to determine which characters can begin and/or end a line when kinsoku line breaking rules are enabled using the kinsoku element (Part 1, §17.3.1.16).

Typically, the characters used to determine which characters shall not end a line are those listed by the kinsoku element in the paragraph properties subclause of this document. This element, when present with a val attribute value of true (or equivalent), specifies that the following settings shall be used instead (for brevity, only those settings which are different are listed below):

Chinese (Simplified)

- Cannot start a line:
  !),.:;?]}¨·ˇ¨–‖’”…:、。〃々〉》」』】〕〗！＂＇），．：；？］｀｜｝～¢
- Cannot end a line:
  ([{·‘“〈《「『【〔〖（．［｛£¥

Chinese (Traditional)

- Cannot start a line:
  !),.:;?]}¢·——’”•…‥·’- 、。〉》」』】〕〞：‖｛ ︶︸︷︺︽︾︿﹀﹂﹄﹃﹁﹃﹂﹄ ﹏·:;?!)}}！），．：；？｜｝、

Korean

- Cannot end a line:
  ([\{£¥‘“〈《「『【〔$（［｛₩

[*Example*: Consider a line of text in a WordprocessingML document within a paragraph marked as Chinese (Simplified) which begins with a % symbol, as follows:

%…

Typically, the kinsoku settings for Chinese (Simplified) do not allow this character to begin a line, so the character before that symbol would be moved down onto this line:

[(%…

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:useAltKinsokuLineBreakRules />
</w:compat>
```

Then the alternate kinsoku rules are in place, which do not prevent the % character from beginning the new line, resulting in the following output:

%…

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.49 useAnsiKerningPairs (Use ANSI Kerning Pairs from Fonts)

This element specifies whether applications shall use the ANSI or Unicode kerning pair information from fonts stored in the document when displaying those characters within the document's contents.

Typically, applications shall use the Unicode kerning pair information in order to determine all possible kerning pairs in the fonts in use. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that the ANSI kerning information shall be used instead.

[*Example*: Consider a WordprocessingML document with text that contains one or more kerning pairs.

If this compatibility setting is turned on:

```
<w:compat>
  <w:useAnsiKerningPairs />
</w:compat>
```

Then the ANSI kerning pairs are used in place of the Unicode kerning pairs, potentially resulting in different line breaks.

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.50 useFELayout (Do Not Bypass East Asian/Complex Script Layout Code)

This element specifies that applications shall not bypass code relating to the layout of East Asian and/or Complex Script characters when presenting this document.

[*Guidance*: Previous word processing applications relied on this flag to determine whether to perform functions which allow for the correct layout of East Asian and Complex Script text. Although current applications no longer rely on this flag (as they should correctly use the Unicode subranges and code pages of the text in use), this flag

should be output in order to ensure that files with this content can be viewed correctly in previous word processors. *end guidance*]

[*Example*: Consider a WordprocessingML document with East Asian text.

If this compatibility setting is turned on:

```
<w:compat>
  <w:useFELayout />
</w:compat>
```

Then the flag is set telling previous applications that East Asian content is present, and they should display the document accordingly. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.51 useNormalStyleForList (Do Not Automatically Apply List Paragraph Style To Bulleted/Numbered Text)

This element specifies whether applications shall automatically apply the paragraph style with the styleId attribute `ListParagraph` when numbering is applied to a paragraph currently formatted using the default paragraph style.

Typically, when a paragraph is formatted using the default paragraph style, and numbering is subsequently applied, the paragraph style with the styleId attribute `ListParagraph` when numbering is applied to ensure that paragraph properties are appropriate for a numbered paragraph. This element, when present with a val attribute value of `true` (or equivalent), specifies that no alternate paragraph style shall ever be applied

[*Example*: Consider a WordprocessingML document with five unnumbered paragraphs:

Example text.

Example text.

Example text.

Example text.

Example text.

If numbering is applied to the three center paragraphs, the default presentation would have the `ListParagraph` style applied as well:

Example text.

- Example text.
- Example text.
- Example text.

Example text.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:useNormalStyleForList />
</w:compat>
```

Then the new paragraph style must not be applied, resulting in the following output:

Example text.

- Example text.
- Example text.
- Example text.

Example text.

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.52 usePrinterMetrics (Use Printer Metrics To Display Documents)

This element specifies whether applications shall use the printer metrics of the currently active printer when determining how to display the contents of a WordprocessingML document. *Printer metrics* are printer-specific settings which can be queried to tell an application how and where text shall be displayed on a printed page.

Typically, applications display the content of a document in a device independent manner - the application is therefore not changing the layout of a document based on the currently attached printer, and instead shall dictate to the printer where characters shall be presented on the page when printed. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that the metrics of the current printer shall be used to display the document instead.

Specifically, when this setting is enabled, the printer metrics are used to determine the number of pixels per logical inch along the screen width and height. This should then be used to compute the pixel height of the fonts requested when displaying the document, as well as to scale between any logical units within the document

(e.g. drawing object sizes) to the appropriate device units. Those units would then need to be scaled back into screen units for final display to a screen, but not scaled again when displayed to a printer.

[*Note*: On the Windows platform, you can use the `GetDeviceCaps` function to retrieve device-specific information for the specified printer. For this specific setting, you can use `GetDeviceCaps(hdc, LOGPIXELSX)` and `GetDeviceCaps(hdc, LOGPIXELSY)` with a printer DC to retrieve the number of pixels per logical inch along the screen width and height. With this, you can then use those DPI metrics to compute a pixel value for the font request in the LOGFONT structure (the LOGFONT structure defines the attributes of a font). A common formula to do this is $S_{px} = S_{pts} * \frac{LOGPIXELSY}{72}$. *end note*]

[*Example*: Consider a WordprocessingML document. The default must use device-independent layout to present the contents of the page.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:usePrinterMetrics />
</w:compat>
```

Then the printer metrics of the current active printer must be used to determine the display of the contents of the document instead, as needed. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

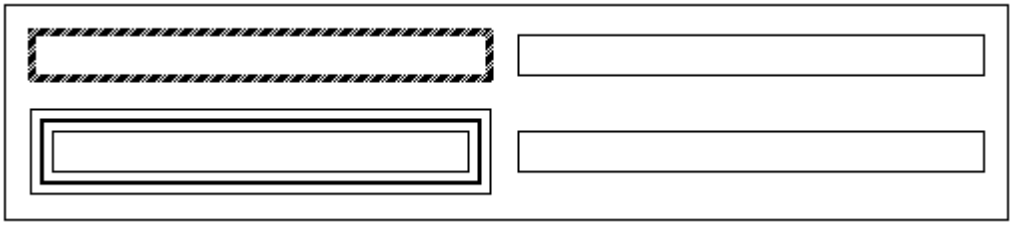
14.7.3.53 useSingleBorderforContiguousCells (Use Simplified Rules For Table Border Conflicts)

This element specifies whether applications should use an alternate simplified algorithm when handling conflicts between adjacent table borders within a table.

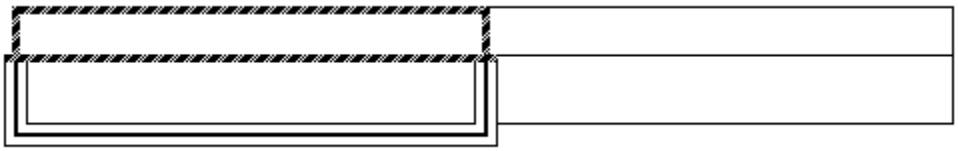
Typically, the conflicts between two adjacent table borders are handled using the conflict resolution algorithm defined in Part 1, §17.4.39 of ISO/IEC 29500. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that rather than using that algorithm to determine the outcome of the conflict to two adjacent borders, that the following logic shall be used instead:

- Cell borders shall supersede table borders
- Cell borders to the right shall supersede cell borders to the left (i.e. the rightmost border wins in conflicts between vertical borders)
- Cell borders below shall supersede cell borders above (i.e. the bottommost border wins in conflicts between horizontal borders)

[*Example*: Consider a WordprocessingML document with cell and table borders defined as follows. In the image below, 0.1" of padding has been added between each cell temporarily to clearly illustrate the borders on each cell and on the table:



The default presentation would have the border conflicts resolved using the algorithm defined by ISO/IEC 29500, resulting in the following table:



However, if this compatibility setting is turned on:

```
<w:compat>
  <w:useSingleBorderforContiguousCells />
</w:compat>
```

Then the simplified table algorithm above shall be used instead (bottom and right cell borders always win), resulting in the following output:

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.54 useWord2002TableStyleRules (Incorrectly Display Top Border of Conditional Columns)

This element specifies whether applications should incorrectly calculate the top border of conditional columns (as specified by a tblStylePr element (Part 1, §17.7.6.6) with a type attribute value of `firstCol`, `lastCol`, `band1Vert`, or `band2Vert`) under the following conditions:

- A conditional formatting has also been defined for the first row (a tblStylePr element with a type attribute of `firstRow`)
- That conditional formatting as been applied to the table using the tblLook element (Part 1, §17.4.56)

Typically, table styles are applied according to the logic defined in Part 1, §17.7.2. This element, when present with a val attribute value of true (or equivalent), specifies that the top border of those conditionally formatted columns should instead be displayed as the top border of the following row.

[*Example*: Consider a WordprocessingML document with table style that defines two conditional formats:

The first column has a one point border
The first row has red shading

That style would be defined as follows:

```
<w:style w:type="table" w:customStyle="1" w:styleId="TableTest">
  <w:name w:val="CompatibilitySetting"/>
  <w:tblStylePr w:type="firstRow">
    <w:tcPr>
      <w:shd w:val="clear" w:color="auto" w:fill="FF0000"/>
    </w:tcPr>
  </w:tblStylePr>
  <w:tblStylePr w:type="firstCol">
    <w:tcPr>
      <w:tcBorders>
        <w:top w:val="single" w:sz="4" w:space="0" w:color="auto"/>
        <w:left w:val="single" w:sz="4" w:space="0" w:color="auto"/>
        <w:bottom w:val="single" w:sz="4" w:space="0" w:color="auto"/>
        <w:right w:val="single" w:sz="4" w:space="0" w:color="auto"/>
      </w:tcBorders>
    </w:tcPr>
  </w:tblStylePr>
</w:style>
```

If the first column and first row formatting is applied, the table would appear as follows:

1,1	1,2
2,1	2,2

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:useWord2002TableStyleRules />
</w:compat>
```

Then the condition described by this element causes the top border defined by the conditional format for the first column to be displayed as the top border for the second column, resulting in the following output:

1,1	1,2
2,1	2,2

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.55 useWord97LineBreakRules (Use Incorrect Inter-Character Spacing Rules)

This element specifies that applications should perform specific calculations (detailed below) when determining inter-character spacing under certain conditions. These calculations would not normally be considered correct.

Typically, the behaviors specified by the following elements are applied unconditionally:

- The autoSpaceDE (Part 1, §17.3.1.2) and autoSpaceDN (Part 1, §17.3.1.3) elements
- The topLinePunct (Part 1, §17.3.1.43) element

The compatibility element described in this subclause, when present with a val attribute value of true (or equivalent), specifies that applications should ignore the settings listed above in the following scenarios:

1. If an ideographic character and a non-ideographic/numeric character are logically adjacent (ignoring all content which is not within a t element), but separated by a field boundary, i.e.:

   - The first character is within a fldSimple element, but the second is not.
   - The characters are separated by a fldChar element with a fldCharType attribute value of end

   Then any appropriate inter-character spacing should be omitted. [*Note*: Inter-character spacing should still be calculated correctly within the field result. *end note*]

2. If a full-width punctuation character appears at the start of a paragraph which also specifies numbering via the numPr element (Part 1, §17.3.1.19), the compression specified by the topLinePunct element is ignored.

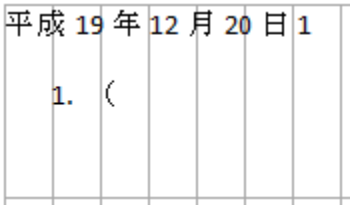
[*Example*: Consider a paragraph which contains a field ending in an ideograph and another paragraph, with numbering, which contains a full-width punctuation character in the first character position:

```
<w:p>
  <w:r>
    <w:fldChar w:fldCharType="begin" />
  </w:r>
  …
  <w:r>
    <w:t>日</w:t>
  </w:r>
  <w:r>
    <w:fldChar w:fldCharType="end" />
```



```
    </w:r>
    <w:r>
      <w:t>1</w:t>
    </w:r>
  </w:p>
  <w:p>
    <w:pPr>
      <w:numPr>
        …
      </w:numPr>
    </w:pPr>
    <w:r>
      <w:t>（</w:t>
    </w:r>
  </w:p>
```

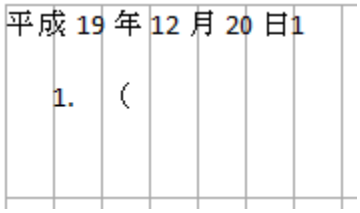
Typically, if both the autoSpaceDN and topLinePunct are `true`, additional spacing is added after the ideograph in the first paragraph and punctuation kerning is applied in the second paragraph (with gridlines added for visual reference):



If this compatibility setting is turned on:

```
  <w:compat>
    <w:useWord97LineBreakRules />
  </w:compat>
```

Then applications should not add any inter-character spacing at th end of the field and should turn off punctuation kerning in the second paragraph:



end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.56 wpJustification (Fit To Expanded Width When Performing Full Justification)

This element specifies that applications should perform a specific algorithm when determining the contents of each line in a fully justified paragraph (resulting from the use of the jc element (Part 1, §**17.3.1.13**)). This setting typically results in more words being fitted into lines (by reducing inter-word spacing as necessary).

Typically, applying full justification to a paragraph does not change the placement of line breaks, as inter-word spacing is expanded to ensure the resulting text is fully justified. This element, when present with a val attribute value of `true` (or equivalent), specifies that applications shall determine the contents of each line in a fully justified paragraph using the following algorithm:

For each line in the fully justified paragraph,

- Determine the actual line width, w, in pixels
- Calculate the "effective" line width by the following factor:

$$w_{\text{effective}} = w_{\text{actual}} + \left(w_{\text{actual}} * \frac{281}{7200}\right)$$

- Determine the text which can be displayed in a line of the "effective" line width
- Decrease the inter-word spacing as necessary to fit that text in the actual line width

[*Example*: Consider a WordprocessingML document with one or more paragraphs using full paragraph justification:

```
<w:p>
  <w:pPr>
    <w:jc w:val="both" />
  </w:pPr>
  …
</w:p>
```

If this compatibility setting is turned on:

```
<w:compat>
  <w:wpJustification />
</w:compat>
```

Then, for a line 1000 pixels wide, an application would calculate the effective width as follows:

$$w_{\text{effective}} = 1000 + \left(1000 * \frac{281}{7200}\right) = 1039\ pixels$$

This effective width is then used to determine how much text can be displayed on line. After calculating the text, the application can display the text on the actual line, fully justified. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.57 wpSpaceWidth (Use Specific Space Width)

(The terms *ascent* and *descent* are used as defined in ISO/IEC 14496-22.)

This element specifies that applications should perform determine the width of the space character for all proportional fonts used in this document using the calculation specified below.

Typically, applications calculate the width of a whitespace character dynamically to optimize for the output device. This element, when present with a val attribute value of `true` (or equivalent), specifies that applications should instead use the following algorithm to determine the width of a whitespace character:

$$w_{\text{space}} = \left(\frac{ascent + descent}{3}\right)$$

where

- w_{space} is the width of a space character
- is the ascent for the font
- is the descent for the font

[*Example*: Consider a WordprocessingML document with this compatibility setting turned on:

```
<w:compat>
  <w:suppressTopSpacingWP />
</w:compat>
```

If the font applied to a run specified an ascent value of 8 points and a descent value of 2 points, each space in that run would have a width of three and one-third points. *end example*]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.3.58 wrapTrailSpaces (Line Wrap Trailing Spaces)

This element specifies whether applications shall perform line wrapping on trailing spaces in the contents of a line when displaying in it a paragraph. *Trailing spaces* are all space characters which are not followed by non-space characters on the same line.

Typically, applications do not line wrap trailing spaces, instead allowing an unbounded number of trailing spaces on a line, with the next non-space character starting at the first character position on the next line. This element, when present with a val attribute value of `true` (or equivalent), specifies that all characters, including trailing spaces, shall be line wrapped normally.

[*Example*: Consider a WordprocessingML document with the following paragraph of text, including a long interstitial of spaces which become trailing spaces when the paragraph is displayed:


```
<w:r>
  <w:t> This is some text followed by a long number of trailing spaces.


                And some more text.</w:t>
</w:r>
```

The default presentation would not wrap those trailing spaces, so the text at the end of the run would begin at the first character position on the second line:

This is some text followed by a long number of trailing spaces.
And some more text.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:wrapTrailSpaces />
</w:compat>
```

Then all trailing spaces would be handled as regular characters when line wrapping, resulting in the following output:

This is some text followed by a long number of trailing spaces.
And some more text.

end example]

This element's content model is defined by the common boolean property definition in Part 1, §17.17.4.

14.7.4 Web Page Settings

14.7.4.1 relyOnVML (Utilize VML When Saving as Web Page)

This element specifies whether applications can utilize the Vector Markup Language format when saving the content of this WordprocessingML document as a web page, when graphical elements that can leverage this format are present in the document.

If this element is omitted, then a graphic image format should be used either in place of or in concert with the Vector Markup Language output in order to specify the formatting and positioning for objects that are part of the resulting web page.

[*Note*: This setting is intended for applications to save web pages that can be supported by legacy web browsers that do not support Vector Markup Language when attempting to read and display the resulting web page. *end note*]

[*Example*: Consider a WordprocessingML document that contains the following content within the web settings part:

```
<w:webSettings>
  <w:relyOnVML w:val="false" />
</w:webSettings>
```

The relyOnVML element has a val attribute value of `false`, which specifies that applications should utilize a graphical image version of all objects that could utilize Vector Markup Language output. This does not preclude the use of the VML output, but does specify that a graphical element must be included as well. *end example*]

This element's content model is defined by the common boolean property definition in Part 1 §17.17.4.

14.8 Miscellaneous Topics

14.8.1 Text Box Content

All VML-based drawing objects (except for connectors) support the addition of rich WordprocessingML content within their extents. When WordprocessingML contents have been added to a VML drawing object, the resulting text is contained within a *text box*.

When WordprocessingML content is contained within a text box, it is allowed within the object by specifying the VML textbox element (§19.1.2.22), which contains within it a single txbxContent element that contains all of the desired WordprocessingML content.

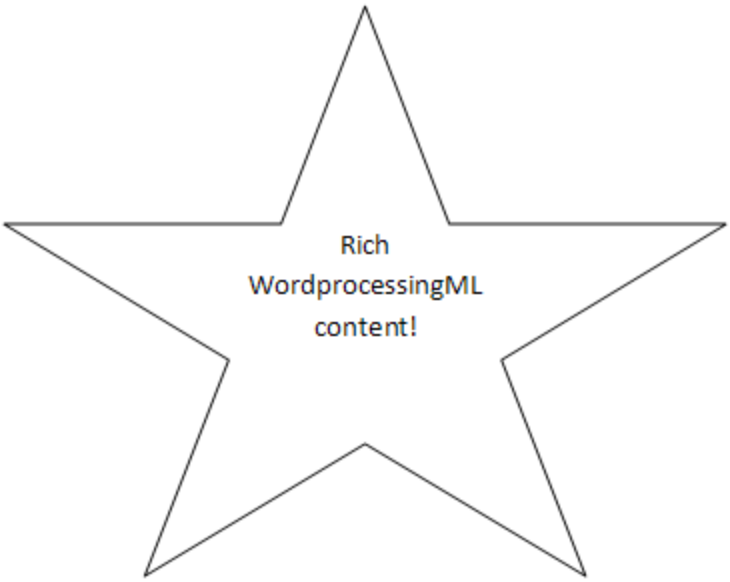
14.8.1.1 txbxContent (Rich Text Box Content Container)

This element specifies that its contents shall be any rich WordprocessingML content, and that this content is the rich contents of a drawing object defined using the Vector Markup Language (VML) syntax (§19.1).

If this element contains within any of its contents any of the following content, then the document shall be considered non-conformant:

- References to other WordprocessingML document stories (comments, footnotes, endnotes)
- Additional txbxContent elements (as part of nested VML objects)

[*Example*: Consider a WordprocessingML document consisting of a single VML shape element (§19.1.2.19) (in this case, a star) that contains within it some WordprocessingML content:



That drawing object now contains a text box, and so uses the syntax for that text box:

```
<v:shape id="_x0000_s1026" type="#_x0000_t12" style="…">
  <v:textbox>
    <w:txbxContent>
      <w:p>
        <w:pPr>
          <w:jc w:val="center"/>
        </w:pPr>
        <w:r>
          <w:t>Rich WordprocessingML content!</w:t>
        </w:r>
      </w:p>
    </w:txbxContent>
  </v:textbox>
</v:shape>
```

end example]

[*Note*: The txbxContent element is the container for the WordprocessingML contained within the text box inside that shape - once inside this element any content (subject to the restrictions defined above) can be used. For compatibility with existing implementations, unqualified elements are used inside the txbxContent element to represent HTML fragments. *end note*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TxbxContent) is located in §A.1. *end note*]

14.9 Fields and Hyperlinks

14.9.1 Syntax

This subclause modifies the fields grammar defined in Part 1, §17.16.1 as necessary to support transitional use of fields.

The syntax rules in this subclause follow the system shown in ISO/IEC 14977: literal text is surrounded by double-quotes (or by apostrophes); the left-square-bracket and right-square-bracket designate the start and end of an option; the left-curly-bracket and right-curly-bracket designate the start and end of a sequence of one-or-more items; the vertical-line indicates an alternative; and each rule ends with a semicolon. Whenever hyphen is used as the exception-symbol (as per ISO/IEC 14977), it is surrounded by white space, and further clarified by a comment.

```
field-type=
   date-and-time  |
   document-automation  |
   document-information  |
   document-property  |
   equations-and-formulas  |
   index-and-tables  |
   links-and-references  |
   mail-merge  |
   numbering  |
   user-information  |
   form-field  |
   user-defined-field |
   transitional-fields ;
transitional-fields=
   "AUTONUM"  |
   "AUTONUMLGL"  |
   "AUTONUMOUT"  |
   "BARCODE",  field-argument  |
   "BIDIOUTLINE"  |
   "EQ",  [switches],  "(",  [eq-argument-list],  ")" (* and see §14.9.4.6 *) |
   "INFO",  info-category,  [field-argument]  ;
eq-argument-list=
   expression,  {eq-list-separator,  expression}  ;
eq-list-separator=
   comma | semicolon  ;        (* depending on the rules in §14.9.4.6 *)
```



```
info-category:
   "AUTHOR"  |  "COMMENTS"  |  "CREATEDATE"  |  "EDITTIME"  |
   "FILENAME"  |  "FILESIZE"  |  "KEYWORDS"  |  "LASTSAVEDBY"  |
   "NUMCHARS"  |  "NUMPAGES"  |  "NUMWORDS"  |  "PRINTDATE"  |
   "REVNUM"  |  "SAVEDATE"  |  "SUBJECT"  |  "TEMPLATE"  |  "TITLE"  ;
```

14.9.2 Legacy language references

Whenever a field requires a language identifier as one of its *field-switches*, that language identifier should be provided using the syntax defined by the ST_Lang simple type (Part 1, §22.9.2.6). However, there exists a legacy mechanism by which language identifiers can be stored. For that mechanism, the following table lists those language codes and their corresponding languages.

This mechanism can be used within the following *field-switches*:

- ADDRESSBLOCK (Part 1, §17.16.5.1), `\l` switch
- BIBLIOGRAPHY (Part 1, §17.16.5.7), `\l` and `\f` switches
- CITATION (Part 1, §17.16.5.8), `\l` switch
- GREETINGLINE (Part 1, §17.16.5.24), `\l` switch
- INDEX (Part 1, §17.16.5.29), `\z` switch

[*Rationale*: This list is maintained for compatibility with documents containing these values. The use of these identifiers is discouraged. *end rationale*]

[*Note*: The second column "Description" is informative only, and is provided as an aid to implementers. Note also that the inclusion of country subtags in the BCP 47 codes makes no assertion about the relationship between nations and languages. Rather, it reflects the historical commercial process by which office software products were localized for some particular market. For example, the Swahili language is spoken in several Eastern African nations. However, the localization identified by the legacy language code `1089` reflected work done in Kenya to address the needs of Swahili users there and thus is mapped to the modern BCP 47 code `sw-KE`. *end note*]

Language Code	Description (informative)	BCP 47 Code
1025	Arabic - Saudi Arabia	ar-SA
1026	Bulgarian	bg-BG
1027	Catalan	ca-ES
1028	Chinese - Taiwan	zh-TW
1029	Czech	cs-CZ
1030	Danish	da-DK
1031	German - Germany	de-DE
1032	Greek	el-GR
1033	English - United States	en-US

Language Code	Description (informative)	BCP 47 Code
1034	Spanish - Spain (Traditional Sort)	es-ES
1035	Finnish	fi-FI
1036	French - France	fr-FR
1037	Hebrew	he-IL
1038	Hungarian	hu-HU
1039	Icelandic	is-IS
1040	Italian - Italy	it-IT
1041	Japanese	ja-JP
1042	Korean	ko-KR
1043	Dutch - Netherlands	nl-NL
1044	Norwegian (Bokmål)	nb-NO
1045	Polish	pl-PL
1046	Portuguese - Brazil	pt-BR
1047	Rhaeto-Romanic	rm-CH
1048	Romanian	ro-RO
1049	Russian	ru-RU
1050	Croatian	hr-HR
1051	Slovak	sk-SK
1052	Albanian - Albania	sq-AL
1053	Swedish	sv-SE
1054	Thai	th-TH
1055	Turkish	tr-TR
1056	Urdu - Pakistan	ur-PK
1057	Indonesian	id-ID
1058	Ukrainian	uk-UA
1059	Belarusian	be-BY
1060	Slovenian	sl-SI
1061	Estonian	et-EE
1062	Latvian	lv-LV
1063	Lithuanian	lt-LT
1064	Tajik	tg-Cyrl-TJ
1065	Farsi	fa-IR
1066	Vietnamese	vi-VN
1067	Armenian - Armenia	hy-AM

Language Code	Description (informative)	BCP 47 Code
1068	Azeri (Latin)	az-Latn-AZ
1069	Basque	eu-ES
1070	Sorbian	wen-DE
1071	FYRO Macedonian	mk-MK
1072	Sutu	st-ZA
1073	Tsonga	ts-ZA
1074	Tswana	tn-ZA
1075	Venda	ven-ZA
1076	Xhosa	xh-ZA
1077	Zulu	zu-ZA
1078	Afrikaans - South Africa	af-ZA
1079	Georgian	ka-GE
1080	Faroese	fo-FO
1081	Hindi	hi-IN
1082	Maltese	mt-MT
1083	Sami	se-NO
1084	Gaelic (Scotland)	gd-GB
1085	Yiddish	yi
1086	Malay - Malaysia	ms-MY
1087	Kazakh	kk-KZ
1088	Kyrgyz (Cyrillic)	ky-KG
1089	Swahili	sw-KE
1090	Turkmen	tk-TM
1091	Uzbek (Latin)	uz-Latn-UZ
1092	Tatar	tt-RU
1093	Bengali (India)	bn-IN
1094	Punjabi	pa-IN
1095	Gujarati	gu-IN
1096	Oriya	or-IN
1097	Tamil	ta-IN
1098	Telugu	te-IN
1099	Kannada	kn-IN
1100	Malayalam	ml-IN
1101	Assamese	as-IN

Language Code	Description (informative)	BCP 47 Code
1102	Marathi	mr-IN
1103	Sanskrit	sa-IN
1104	Mongolian (Cyrillic)	mn-MN
1105	Tibetan - People's Republic of China	bo-CN
1106	Welsh	cy-GB
1107	Khmer	km-KH
1108	Lao	lo-LA
1109	Burmese	my-MM
1110	Galician	gl-ES
1111	Konkani	kok-IN
1112	Manipuri	mni
1113	Sindhi - India	sd-IN
1114	Syriac	syr-SY
1115	Sinhalese - Sri Lanka	si-LK
1116	Cherokee - United States	chr-US
1117	Inuktitut	iu-Cans-CA
1118	Amharic - Ethiopia	am-ET
1119	Tamazight (Arabic)	tmz
1120	Kashmiri (Arabic)	ks-Arab-IN
1121	Nepali	ne-NP
1122	Frisian - Netherlands	fy-NL
1123	Pashto	ps-AF
1124	Filipino	fil-PH
1125	Divehi	dv-MV
1126	Edo	bin-NG
1127	Fulfulde - Nigeria	fuv-NG
1128	Hausa - Nigeria	ha-Latn-NG
1129	Ibibio - Nigeria	ibb-NG
1130	Yoruba	yo-NG
1131	Quecha - Bolivia	quz-BO
1132	Sepedi	nso-ZA
1136	Igbo - Nigeria	ig-NG
1137	Kanuri - Nigeria	kr-NG
1138	Oromo	gaz-ET

Language Code	Description (informative)	BCP 47 Code
1139	Tigrigna - Ethiopia	ti-ER
1140	Guarani - Paraguay	gn-PY
1141	Hawaiian - United States	haw-US
1142	Latin	la
1143	Somali	so-SO
1144	Yi	ii-CN
1145	Papiamentu	pap-AN
1152	Uighur - China	ug-Arab-CN
1153	Maori - New Zealand	mi-NZ
2049	Arabic - Iraq	ar-IQ
2052	Chinese - People's Republic of China	zh-CN
2055	German - Switzerland	de-CH
2057	English - United Kingdom	en-GB
2058	Spanish - Mexico	es-MX
2060	French - Belgium	fr-BE
2064	Italian - Switzerland	it-CH
2067	Dutch - Belgium	nl-BE
2068	Norwegian (Nynorsk)	nn-NO
2070	Portuguese - Portugal	pt-PT
2072	Romanian - Moldava	ro-MO
2073	Russian - Moldava	ru-MO
2074	Serbian (Latin)	sr-Latn-CS
2077	Swedish - Finland	sv-FI
2080	Urdu - India	ur-IN
2092	Azeri (Cyrillic)	az-Cyrl-AZ
2108	Gaelic (Ireland)	ga-IE
2110	Malay - Brunei Darussalam	ms-BN
2115	Uzbek (Cyrillic)	uz-Cyrl-UZ
2117	Bengali (Bangladesh)	bn-BD
2118	Punjabi (Pakistan)	pa-PK
2128	Mongolian (Mongolian)	mn-Mong-CN
2129	Tibetan - Bhutan	bo-BT
2137	Sindhi - Pakistan	sd-PK
2143	Tamazight (Latin)	tzm-Latn-DZ

Language Code	Description (informative)	BCP 47 Code
2144	Kashmiri (Devanagari)	ks-Deva-IN
2145	Nepali - India	ne-IN
2155	Quecha - Ecuador	quz-EC
2163	Tigrigna - Eritrea	ti-ET
3073	Arabic - Egypt	ar-EG
3076	Chinese - Hong Kong SAR	zh-HK
3079	German - Austria	de-AT
3081	English - Australia	en-AU
3082	Spanish - Spain (Modern Sort)	es-ES
3084	French - Canada	fr-CA
3098	Serbian (Cyrillic)	sr-Cyrl-CS
3179	Quecha - Peru	quz-PE
4097	Arabic - Libya	ar-LY
4100	Chinese - Singapore	zh-SG
4103	German - Luxembourg	de-LU
4105	English - Canada	en-CA
4106	Spanish - Guatemala	es-GT
4108	French - Switzerland	fr-CH
4122	Croatian (Bosnia/Herzegovina)	hr-BA
5121	Arabic - Algeria	ar-DZ
5124	Chinese - Macao SAR	zh-MO
5127	German - Liechtenstein	de-LI
5129	English - New Zealand	en-NZ
5130	Spanish - Costa Rica	es-CR
5132	French - Luxembourg	fr-LU
5146	Bosnian (Bosnia/Herzegovina)	bs-Latn-BA
6145	Arabic - Morocco	ar-MO
6153	English - Ireland	en-IE
6154	Spanish - Panama	es-PA
6156	French - Monaco	fr-MC
7169	Arabic - Tunisia	ar-TN
7177	English - South Africa	en-ZA
7178	Spanish - Dominican Republic	es-DO
7180	French - West Indies	fr-029

Language Code	Description (informative)	BCP 47 Code
8193	Arabic - Oman	ar-OM
8201	English - Jamaica	en-JM
8202	Spanish - Venezuela	es-VE
8204	French - Reunion	fr-RE
9217	Arabic - Yemen	ar-YE
9225	English - Caribbean	en-029
9226	Spanish - Colombia	es-CO
9228	French - Democratic Rep. of Congo	fr-CG
10241	Arabic - Syria	ar-SY
10249	English - Belize	en-BZ
10250	Spanish - Peru	es-PE
10252	French - Senegal	fr-SN
11265	Arabic - Jordan	ar-JO
11273	English - Trinidad	en-TT
11274	Spanish - Argentina	es-AR
11276	French - Cameroon	fr-CM
12289	Arabic - Lebanon	ar-LB
12297	English - Zimbabwe	en-ZW
12298	Spanish - Ecuador	es-EC
12300	French - Cote d'Ivoire	fr-CI
13313	Arabic - Kuwait	ar-KW
13321	English - Philippines	en-PH
13322	Spanish - Chile	es-CL
13324	French - Mali	fr-ML
14337	Arabic - U.A.E.	ar-AE
14345	English - Indonesia	en-ID
14346	Spanish - Uruguay	es-UY
14348	French - Morocco	fr-MA
15361	Arabic - Bahrain	ar-BH
15369	English - Hong Kong SAR	en-HK
15370	Spanish - Paraguay	es-PY
15372	French - Haiti	fr-HT
16385	Arabic - Qatar	ar-QA
16393	English - India	en-IN

Language Code	Description (informative)	BCP 47 Code
16394	Spanish - Bolivia	es-BO
17417	English - Malaysia	en-MY
17418	Spanish - El Salvador	es-SV
18441	English - Singapore	en-SG
18442	Spanish - Honduras	es-HN
19466	Spanish - Nicaragua	es-NI
20490	Spanish - Puerto Rico	es-PR
21514	Spanish - United States	es-US
58378	Spanish - Latin America	es-419
58380	French - North Africa	fr-015
Any other value	Undefined. Shall not be used.	

14.9.3 Use of DOS File Paths

The following fields allow the use of a DOS file path in place of the (preferred) IRI syntax:

- INCLUDEPICTURE (Part 1, §17.16.5.27)
- INCLUDETEXT (Part 1, §17.16.5.28)

When a DOS file path is specified in a *field-argument*, each backslash character shall be preceded directly by another backslash character [*Example*: E:\\example.docx *end example*] If *field-argument* contains white space, it shall be enclosed in double quotes.

14.9.4 Field definitions

14.9.4.1 AUTONUM

Syntax:

AUTONUM [*switches*]

Description: In paragraphs formatted with one of the nine built-in heading styles, paragraph numbering restarts at 1 in each successive heading level. If headings that contain AUTONUM fields are followed by body text paragraphs that also contain AUTONUM fields, the paragraph numbering of the body text is restarted at 1 after each heading. If the headings don't contain AUTONUM fields, body text paragraphs that contain AUTONUM fields are numbered in a continuous, sequential series throughout the document. [*Note*: This field is supported for legacy reasons, It is recommended that LISTNUM (Part 1, §17.16.5.33) be used instead. *end note*]

The XML generated for a complex field implementation shall not have the optional field value stored.

Field Value: A new paragraph number in ascending sequential order.

Switches: Zero or one of the *general-formatting-switch*es, or zero or more of the following *field-specific-switch*es.

`\s` *field-argument*	*text* in this switch's *field-argument* specifies the separator character to be used. If `\s` is omitted, a period (.) is used.

[*Example*: When the following fields are updated:

```
AUTONUM
AUTONUM \* Arabic \s :
AUTONUM \* alphabetic \s " "xxx
AUTONUM \* ROMAN
AUTONUM \* OrdText
```

The results are:

```
1.
2:
c xxx
IV.
fifth.
```

end example]

14.9.4.2 AUTONUMLGL

Syntax:

AUTONUMLGL [*switches*]

Description: For legal and technical publications, use the nine built-in heading styles to format headings in the document, and then insert an `AUTONUMLGL` field at the beginning of each heading paragraph. The numbers reflect the heading levels that correspond to the heading styles. If an `AUTONUMLGL` field is inserted in paragraphs of body text paragraphs not formatted with built-in heading styles, the number of the preceding heading is included in the paragraph number. [*Note*: This field is supported for legacy reasons, It is recommended that `LISTNUM` (Part 1, §17.16.5.33) be used instead. *end note*]

This field only makes sense in terms of multi-level headings. Given the following headings:

Heading 1
Heading 2
Heading 2
Heading 1

this field allows

1. Heading 1
1.1. Heading 2
1.2. Heading 2
2. Heading 1

At each level, the numbering sequence does two things—it increments specific to that level, and it includes the value from the previous level.

The XML generated for a complex field implementation shall not have the optional field value stored.

Field Value: A new paragraph number in ascending sequential order.

Switches: Zero or one of the *general-formatting-switch*es, or zero or more of the following *field-specific-switch*es.

`\e`	Removes the trailing separator (period).
`\s` *field-argument*	*text* in this switch's *field-argument* specifies the separator character to be used. If `\s` is omitted, a period (.) is used.

[*Example*: When the following fields are updated:

```
AUTONUMLGL
AUTONUMLGL \* Arabic \s :
AUTONUMLGL \* alphabetic \s " "xxx
AUTONUMLGL \* ROMAN
AUTONUMLGL \e xxx
```

The results are:

```
1.
2:
c xxx
IV.
5xxx
```

end example]

14.9.4.3 AUTONUMOUT

Syntax:

AUTONUMOUT

Description: Use the nine built-in heading styles to format headings in the document, and then insert an AUTONUMOUT field at the beginning of each heading paragraph. The numbers reflect the heading levels that

correspond to the heading styles. [*Note*: This field is supported for legacy reasons, It is recommended that `LISTNUM` (Part 1, §17.16.5.33) be used instead. *end note*]

The XML generated for a complex field implementation shall not have the optional field value stored.

This field allows the numbering to be incremented based on the heading level. Given the following:

`{AutoNumOut}` Heading 1
`{AutoNumOut}` Heading 2
`{AutoNumOut}` Heading 2
`{AutoNumOut}` Heading 1

results in

I. Heading 1
A. Heading 2
B. Heading 2
II. Heading 1

Field Value: A paragraph number.

Switches: None.

[*Example*: When the following fields are updated:

```
AUTONUMOUT
AUTONUMOUT
```

The results are:

```
1.
2.
```

end example]

14.9.4.4 BARCODE

Syntax:

BARCODE *field-argument* [*switches*]

Description: Produces a postal bar code in a machine-readable form of address used by the U.S. Postal Service. The barcode is in the form of either a POSTNET delivery-point bar code or a Facing Identification Mark (FIM). *text* in *field-argument* can be either a postal address or a bookmark name. In the case of a postal address, all that is needed is a 5-digit or 9-digit ZIP code; the rest of the address is superfluous.

Field Value: A postal bar code.

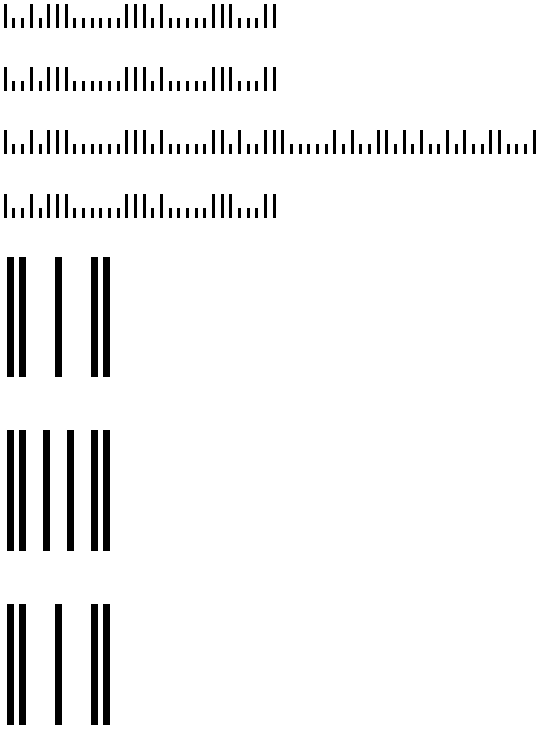
Switches: Zero or more of the following *field-specific-switch*es.

`\b`	Indicates that *text* in *field-argument* is the name of a bookmark.
`\f` *field-argument*	Inserts a Facing Identification Mark (FIM). *text* in this switch's *field-argument* shall be either "A" (courtesy reply mark) or "C" (business reply mark).
`\u`	Indicates that *text* in *field-argument* is a U.S. postal address.

[*Example*: Consider the case in which `PostalAddress` is the name of a bookmark for the text "2051 Swans Neck Way, Reston VA 20191". When the following fields are updated:

```
BARCODE 20191
BARCODE 20191 \u
BARCODE 20191-4023 \u
BARCODE "2051 Swans Neck Way, Reston VA 20191" \u
BARCODE "2051 Swans Neck Way, Reston VA 20191" \f A
BARCODE 20191 \f C
BARCODE PostalAddress \b \f A
```

The results are:



end example]

14.9.4.5 BIDIOUTLINE

Syntax:

```
BIDIOUTLINE
```

Description: This field is identical to the AUTONUMLGL field (§14.9.4.3), except for the separator that delimits each level of the paragraph numbering (this field uses a hyphen-minus (U+002D) instead of a full stop (U+002E) character as the default separator character).

Field Value: A new paragraph number in ascending sequential order, as defined by the description in §14.9.4.3.

Switches: None.

14.9.4.6 EQ

Syntax:

EQ [*switches*] (*eq-argument-list*) [*switches*]

eq-argument-list is a list of arguments separated using a separator character. For implementations using a period (.) as the radix point, the separator character is a comma (,). For implementations using a comma (,) as the radix point, the separator character is a semicolon (;).

Description: Computes the specified mathematical equation.

Field Value: The result of the specified mathematical equation. [*Note*: The result of an EQ field can be used as an argument in another EQ field's *eq-argument-list*. *end note*]

Switches: The left-hand *switch*es can only be one of the following: \a, \b, \d, \f, \i, \l, \o, \r, \s, and \x. Each of these switches has one or more subswitches, as shown below.

\a produces an array using the argument values in *eq-argument-list* (which are in row-major order) and the *field-specific-switch*es below:

\ac	Alignment is centered in each array column.
\al	Alignment is left in each array column.
\ar	Alignment is right in each array column.
\co *field-argument*	The number of columns in the array is specified by *text* in this switch's *field-argument*. In the absence of this switch, the number is 1.
\hs *field-argument*	Adds the integral number of points of horizontal spacing specified by *text* in this switch's *field-argument* between columns.
\vs *field-argument*	Adds the integral number of points of vertical spacing specified by *text* in this switch's *field-argument* between lines.

`\b` brackets the single element in *eq-argument-list* in a size appropriate for that element. The default form of brackets is parentheses. The *field-specific-switch*es below can be used:

`\bc` `\`*char*	Uses the character designated by *char* as both the left and right bracket character. However, if *char* is `{`, `[`, `(`, or `<`, that character is used for the left bracket, and `}`, `]`, `)`, or `>`, respectively, is used for right bracket.
`\lc` `\`*char*	Uses the character designated by *char* as the left bracket character.
`\rc` `\`*char*	Uses the character designated by *char* as the right bracket character.

`\d` Controls where the next character following the EQ field is drawn (that is, the displacement). *eq-argument-list* shall have no arguments. The *field-specific-switch*es below can be used:

`\ba` *field-argument*	Draws to the left (backward) the integral number of points specified by *text* in this switch's *field-argument*.
`\fo` *field-argument*	Draws to the right (forward) the integral number of points specified by *text* in this switch's *field-argument*.
`\li`	Underlines the space up to the next character.

`\f` Creates a fraction with the first argument as numerator and the second argument as denominator, centered above and below the division line, respectively. *eq-argument-list* shall have exactly two arguments. There are no *field-specific-switch*es for this switch.

`\i` Creates an integral using the specified or default symbol and three elements. The first argument is the lower limit, the second is the upper limit, and the third is the integrand. *eq-argument-list* shall have exactly three arguments. The *field-specific-switch*es below can be used:

`\fc` `\`*char*	Uses the character designated by *char* as the fixed-height character for the symbol.
`\in`	Uses an inline format with the limits displayed to the right of the symbol instead of above and below it.
`\pr`	Uses the symbol Capital pi and creates a product.
`\su`	Uses the symbol Capital sigma and creates a summation.
`\vc` `\`*char*	Uses the character designated by *char* as the variable-height character for the symbol. The symbol matches the height of the third argument.

`\lf` Creates a list from an arbitrary number of arguments. There are no *field-specific-switch*es for this switch.

`\o` Using an arbitrary number of arguments, displays each successive argument on top of the previous one. Each character is displayed within an invisible character box, with the switches being available to align the boxes on top of one another. The *field-specific-switch*es below can be used:

`\ac`	Alignment character box center (the default).
`\al`	Alignment character box left.
`\ar`	Alignment character box right.

`\r` Creates a radical. *eq-argument-list* shall have either one or two arguments. If it has one argument, the result is the square root of that argument. If it has two arguments, the result is the *n*th root of the second argument, where *n* is the first argument. There are no *field-specific-switch*es for this switch.

`\s` Creates a subscript or superscript. One or more arguments are permitted. If more than one element is specified, the elements are stacked and left-aligned. The *field-specific-switch*es below can be used:

`\ai` *field-argument*	Adds space above a line in a paragraph by the integral number of points specified by *text* in this switch's *field-argument*. The default is 2 points.
`\di` *field-argument*	Adds space below a line in a paragraph by the integral number of points specified by *text* in this switch's *field-argument*.
`\do` *field-argument*	Moves a single argument below the adjacent text by the integral number of points specified by *text* in this switch's *field-argument*. The default is 2 points.
`\up` *field-argument*	Moves a single argument above the adjacent text by the integral number of points specified by *text* in this switch's *field-argument*.

`\x` Creates one or more border segments around a single argument. By default, all four borders are added. *eq-argument-list* shall have no arguments. The *field-specific-switch*es below can be used:

`\bo`	Draws a horizontal border below the argument.
`\le`	Draws a vertical border to the left of the argument.
`\ri`	Draws a vertical border to the right of the argument.
`\to`	Draws a horizontal border above the argument.

[*Example*: When the following fields are updated:

EQ \a \co 2 \ac \hs 10 (1000, 20, A, Sunday)
EQ \b \bc \| (-100) EQ \b \bc \| (\r(3, a + b)
xx EQ \d \fo 20 () xx EQ \d \fo 30 \li ()xx
EQ \f (1, 32) EQ \f (7, 64)
EQ \i (0, ∞, x) EQ \i \su \in (0, 10, x) EQ \i \pr \in (0, 5, x)
EQ \i \fc \{ (0, 5, \f (x, 0.34)) EQ \i \vc \{ (0, 5, \f (x, 0.34))

EQ \l (0, 10)
EQ \b \lc \[\rc \) (\l (0, 10))

EQ \o (0, 0, 0) EQ \o (0, +) EQ \o \ar (0, |, _)
EQ \r (2) EQ \r (2, x)
a EQ \s \up (2) + b EQ \s \up (2)
a EQ \x (+) b a EQ \x \to \le (+) b a EQ \x \bo \ri (+) b

The results are:

1000 20
A Sunday

| -100 | $\left| \sqrt[3]{a + b} \right|$

xx xx_____ xx

$\frac{1}{32}$ $\frac{7}{64}$

$\int_0^\infty x$ $\sum_0^{10} x$ $\prod_0^5 x$

$\{_0^5 \frac{x}{0.34}$ $\left\{_0^5 \frac{x}{0.34}\right.$

0, 10
[0, 10)

$\mathbf{0}$ θ $\underline{\emptyset}$

$\sqrt{2}$ $\sqrt[2]{x}$

$a^2 + b^2$

a $\boxed{+}$ b a $\overline{+}$ b a $\underline{+}$ b

end example]

14.9.4.7 INFO

Syntax:

```
INFO info-category [ field-argument ] [ switches ]
```


This field is documented for purposes of backwards compatibility. Each permitted value for *info-category* is also permitted as a *field-type*. Instances of the INFO field shall be treated as an instance of the *field-type* with the same value as *info-category*; that is, as if the INFO token was not present.

14.9.4.8 QUOTE

This field retrieves the text specified by *text* in *field-argument*. In strict conformance mode, this text may include any other fields except SYMBOL. However, in transitional conformance mode, this text may include any other fields except AUTONUM, AUTONUMLGL, AUTONUMOUT, and SYMBOL.

14.9.5 fldData (Custom Field Data)

This element specifies custom field data which shall be associated with the parent field. No information or semantics are applied to the contents of this data by ISO/IEC 29500, and therefore this field can be used as desired to store additional application-defined data with the field. However, applications should not lose the contents of this custom data if they do not understand or utilize it (i.e. the information should continue to be saved with the file).

If this element is omitted, then no custom field data is stored with the parent field. If the type attribute of the current field character is not `start`, then his setting can be ignored.

[*Example*: Consider the following WordprocessingML fragment for a complex field:

```
<w:r>
  <w:fldChar w:fldCharType="start">
    <w:fldData xml:space="preserve">///3645ERKJHE</w:fldData>
  </w:fldChar>
</w:r>
<w:r>
  <w:instrText>PRIVATE</w:instrText>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="separate" />
</w:r>
…
```

The fldData element contains custom data stored with this `PRIVATE` field (Part 1, §17.16.5.48), the contents of which are determined by a hosting application. *end example*]

Attributes	Description
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.or	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. [*Example*: Consider the following run contained within a WordprocessingML document: `<w:r>`

Attributes	Description

g/XML/1998/namespace	`<w:t> significant whitespace </w:t>`
	`</w:r>`
	
	Although there are three spaces on each side of the text content in the run, that whitespace has not been specifically marked as significant, therefore it is subject to the space preservation rules currently specified in that run's scope. *end example*]
	
	The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Text) is located in §A.1. *end note*]

14.9.6 fldData (Custom Field Data)

This element specifies custom field data which shall be associated with the parent field. No information or semantics are applied to the contents of this data by ISO/IEC 29500, and therefore this field can be used as desired to store additional application-defined data with the field. However, applications should not lose the contents of this custom data if they do not understand or utilize it (i.e. the information should continue to be saved with the file).

If this element is omitted, then no custom field data is stored with the parent field.

[*Example*: Consider the following WordprocessingML fragment for a simple field:

```
<w:fldSimple w:instr="PRIVATE">
  <w:fldData xml:space="preserve">///3645ERKJHE</w:fldData>
</w:fldSimple>
```

The fldData element contains custom data stored with this `PRIVATE` field (Part 1, §17.16.5.48), the contents of which are determined by a hosting application. *end example*]

Attributes	Description

xml:space (Content Contains Significant Whitespace)
 	
Namespace: http://www.w3.org/XML/1998/namespace
	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules.
	
	[*Example*: Consider the following run contained within a WordprocessingML document:
	
	`<w:r>`
	`<w:t> significant whitespace </w:t>`
	`</w:r>`
	
	Although there are three spaces on each side of the text content in the run, that whitespace has not been specifically marked as significant, therefore it is subject to the space preservation rules currently specified in that run's scope. *end example*]
	
	The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Text) is located in §A.1. *end note*]

14.9.7 hyperlink (Hyperlink) (Part 1, §17.16.22)

Attributes	Description
id (Hyperlink Target) Namespace: .../officeDocument/2006/relationships	The same as the id attribute in Part 1, §17.16.22.

14.10 Simple Types

The following additional simple type information in the http://schemas.openxmlformats.org/wordprocessingml/2006/main namespace is used for documents of a transitional conformance class.

14.10.1 Additional member types for the union in ST_DecimalNumberOrPercent (Part 1, §17.18.11)

The value space of the following additional member types can be used within the context of this simple type for a document of a transitional conformance class.

- The ST_UnqualifiedPercentage simple type (§14.10.10).

14.10.2 Additional enumeration values for ST_Jc (Part 1, §17.18.44)

The following addtional enumeration values can be specified for a document of a transitional conformance class.

Enumeration Value	Description
left (Align to Leading Edge)	Semantically equivalent to start.
right (Align to Trailing Edge)	Semantically equivalent to end.

14.10.3 Additional enumeration values for ST_JcTable (Part 1, §17.18.45)

The following addtional enumeration values can be specified for a document of a transitional conformance class.

Enumeration Value	Description
left (Align to Starting Edge)	Specifies that the table shall be aligned to the leading edge of the text flow – the left text margin (for a left-to-right table); or the right text margin (for a right-to-left table) in the document. (See Part 1, §17.4.1)

Enumeration Value	Description
right (Align to Trailing Edge)	Specifies that the table shall be aligned to the trailing edge of the text flow – the right text margin (for a left-to-right table); or the left text margin (for a right-to-left table) in the document. (See Part 1, §17.4.1)

14.10.4 Additional enumeration values for ST_NumberFormat (Part 1, §17.18.59)

Enumeration Value	Description
decimalFullWidth2 (Full Width Arabic Numerals Alternate)	Specifies that the sequence shall consist of a set of full-width Arabic numbering. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–9 and then those same characters are combined with each other and ０ (represents the number zero) to construct the remaining values. The set of characters used by this numbering format for values 0–9 is U+FF10–U+FF19, respectively. For values greater than the size of the set, the number is constructed by following these steps: **1.** Divide the value by 10 and write the symbol which represents the remainder. **2.** Divide the quotient of the previous division by 10 and write the symbol, which represents the remainder, to the left of the existing position. **3.** Repeat step 2 until the remaining value is equal to zero. [*Example*: The numbering for the items should be represented by the following pattern: １, ２, ３, …, ８, ９, １０, １１, １２, …, １８, １９, ２０, ２１, … *end example*]

14.10.5 Additional enumeration values for ST_StyleSort (Part 1, §17.18.82)

The following addtional enumeration values can be specified for a document of a transitional conformance class.

Enumeration Value	Description
0000 (Sort by Style Name)	Specifies that styles which are visible should be sorted by their names.
0001 (Sort by Style Priority)	Specifies that styles which are visible should be sorted by their UI priority using the uiPriority element (Part 1, §**17.7.4.19**).
0002 (Sort by Default Method)	Specifies that styles which are visible should be sorted

Enumeration Value	Description
	by the default sorting of the host application.
0003 (Sort by Font)	Specifies that styles which are visible should be sorted by the font which they apply.
0004 (Sort by Based On Style)	Specifies that styles which are visible should be sorted by the style on which they are based using the basedOn element (Part 1, §**17.7.4.3**).
0005 (Sort by Style Type)	Specifies that styles which are visible should be sorted by their style types (i.e. character, linked, paragraph).

14.10.6 Additional enumeration values for ST_TabJc (Part 1, §17.18.84)

The following addtional enumeration values can be specified for a document of a transitional conformance class.

Enumeration Value	Description
left (Leading Tab)	Semantically equivalent to start.
right (Trailing Tab)	Semantically equivalent to end.

14.10.7 Additional enumeration values for ST_TextDirection (Part 1, §17.18.93)

The following addtional enumeration values can be specified for a document of a transitional conformance class.

Enumeration Value	Description
btLr (Lines Flow From Left to Right)	Semantically equivalent to lr.
lrTb (Lines Flow From Top To Bottom)	Semantically equivalent to tb.
lrTbV (Lines Flow From Top to Bottom, Rotated)	Semantically equivalent to tbV.
tbLrV (Lines Flow From Left to Right, Rotated)	Semantically equivalent to lrV.
tbRl (Lines Flow From Right to Left)	Semantically equivalent to rl.
tbRlV (Lines Flow From Right to Left, Rotated)	Semantically equivalent to rlV.

14.10.8 Additional member types for the union in ST_TextScale (Part 1, §17.18.95)

The value space of the following additional member types can be used within the context of this simple type for a document of a transitional conformance class.

The ST_TextScaleDecimal simple type (§9.10.11).

14.10.9 ST_Cnf (Conditional Formatting Bitmask)

This simple type specifies the format for the set of conditional formatting properties that have been applied to this object.

These properties are expressed using a string serialization of a binary bitmask for each of the following properties (reading from the first character position right):

- First Row - Is this the first row of the table?
- Last Row - Is this the last row of the table?
- First Column - Does this belong to the first column of the table?
- Last Column - Does this belong to the last column of the table?
- Band 1 Vertical - Does this belong to a column which should receive band 1 formatting? This property specifies whether the cell should receive the formatting specified for odd-numbered columns (e.g. 1,3,5,…)
- Band 2 Vertical - Does this belong to a column which should receive band 2 formatting? This property specifies whether the cell should receive the formatting specified for even-numbered columns (e.g. 2,4,6…)
- Band 1 Horizontal - Does this receive band 1 formatting? This property specifies whether the cell should receive the formatting specified for odd-numbered rows (e.g. 1,3,5,…)
- Band 2 Horizontal - Does this receive band 2 formatting? This property specifies whether the cell should receive the formatting specified for even-numbered rows (e.g. 2,4,6…)
- NE Cell - Is this part of the top-right corner of the table?
- NW Cell - Is this part of the top-left corner of the table?
- SE Cell - Is this part of the bottom-right corner of the table?
- SW Cell - Is this part of the bottom-left corner of the table?

For each of these properties, a value of `1` in the specified character position in the string means that the value is true, a value of `0` means false. All values shall be specified.

[*Example*: Consider a paragraph in the top right corner of a table with a table style applied. This paragraph would need to specify the following WordprocessingML:

```
<w:p>
  <w:pPr>
    <w:cnfStyle w:val="101000000100" />
    …
  </w:pPr>
  …
</w:p>
```

This paragraph specifies that it has the conditional properties from the table style for the first column, first row, and the NW corner of the parent table by setting the appropriate bits in the val attribute. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a length of exactly 12 characters.

- This simple type's contents shall match the following regular expression pattern: [01]*.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Cnf) is located in §A.1. *end note*]

14.10.10 ST_UnqualifiedPercentage (Percentage Value Without Percent Sign)

This simple type specifies additional formats for percentage-based values which can only be used within the transitional conformance class.

Specifically, this value allows percentage-based values to be specified as follows:

- For the w attribute in CT_TblWidth (Part 1, §17.4.88), the value is stored in 50ths of a percent.
- For all other uses, the value is stored in whole percentage points.

[*Example*: Consider the following WordprocessingML fragment:

```
<w:tblW w:w="1000" w:type="pct" />
```

The tblW element is based on the CT_TblWidth complex type, and the type attribute's value is `pct`, which means that this value is measured in 50ths of a percent (i.e. `1000` is equal to 20%). *end example*]

This simple type's contents are a restriction of the W3C XML Schema integer datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_UnqualifiedPercentage) is located in §A.1. *end note*]

14.10.11 ST_TextScaleDecimal (Text Expansion/Compression Percentage)

This simple type specifies that the percentage by which the contents of a run shall be expanded or compressed with respect to its normal (100%) character width, with a minimum width of 1% and maximum width of 600%.

[*Example*: Consider a run of text which must be expanded to 300% when displaying each character within the contents of the run. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:w w:val="300"/>
</w:rPr>
```

This run explicitly declares that the w value is `300`, so the contents of this run appear at 300% of their normal character width by expanding the width of each character. *end example*]

This simple type's contents are a restriction of the W3C XML Schema integer datatype.

This simple type also specifies the following restrictions:

This simple type has a minimum value of greater than or equal to 0.
This simple type has a maximum value of less than or equal to 600.

14.11 Changed attributes

The following attributes, which are defined in subclauses within Part 1, §17, “WordprocessingML”, have different source relationships when used in documents of the Transitional conformance class:

14.11.1 Changed attribute for contentPart element (Part 1, §17.3.3.2)

Attributes	Description
id (Relationship to Part) Namespace: …/officeDocument/2006/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: http://schemas.openxmlformats.org/officeDocument/2006/customXml for the contentPart element http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer for the footerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/header for the headerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.2 Changed attribute for control element (Part 1, §17.3.3.3)

Attributes	Description
id (Embedded Control Properties Relationship Reference) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID for the relationship that contains the properties for this embedded control. This property bag is contained in a separate part within the Office Open XML package. The relationship explicitly targeted by this attribute shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/control or the document shall be considered non-conformant. If this attribute is omitted, then the embedded control shall be given no property bag when instantiated. [*Example*: Consider the following WordprocessingML markup for an embedded control in a document: `<w:control r:id="rId5" w:name="CheckBox1" w:shapeid="_x0000_s1027" />` The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rId5` must contain the property data for this embedded control. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.3 Changed attribute for movie element (Part 1, §17.3.3.17)

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: http://schemas.openxmlformats.org/officeDocument/2006/customXml for the contentPart element http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer for the footerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/header for the headerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute:

Attributes	Description
	`<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.4 Changed attribute for objectEmbed element (Part 1, §17.3.3.20)

Attributes	Description
id (Relationship to Embedded Object Data) Namespace: …/officeDocument/2006/relationships	Specifies the relationship ID for the relationship that targets the Embedded Object Part containing the embedded object data. The specified relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/oleObject or the document shall be considered non-conformant. [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` targets the part containing the corresponding embedded object information. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.5 Changed attribute for objectLink element (Part 1, §17.3.3.21)

Attributes	Description
id (Relationship to Embedded Object Data) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID for the relationship that targets the Embedded Object Part containing the embedded object data. The specified relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/oleObject or the document shall be considered non-conformant. [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` targets the part containing the corresponding embedded object information. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.6 Changed attribute for bottom element (Part 1, §17.6.2)

Attributes	Description

bottomLeft (Custom Defined Bottom Left Border Relationship Reference)
 	
Namespace: .../officeDocument/2006/relationships
	Specifies the relationship ID for the relationship that contains the custom bottom left border image for the parent element. This custom border image is contained in a separate part within the WordprocessingML package.
	
	The relationship explicitly targeted by this attribute shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/image or the document shall be considered non-conformant.
	
	If this attribute is omitted, then no custom bottom left border shall be used.
	
	[*Example*: Consider the following WordprocessingML markup for a custom bottom left border in a document:
	
	`<w:bottom w:val="custom"`
	`r:bottomLeft="rIdCustomBottomLeftBorder" …/>`
	
	The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rIdCustomBottomLeftBorder` must contain the custom bottom left border image for the document. *end example*]
	
	The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

bottomRight (Custom Defined	Specifies the relationship ID for the relationship that contains the custom bottom right border image for the parent element. This custom border image is contained in a

Attributes	Description
Bottom Right Border Relationship Reference) Namespace: …/officeDocument/2006/relationships	separate part within the WordprocessingML package. The relationship explicitly targeted by this attribute shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/image or the document shall be considered non-conformant. If this attribute is omitted, then no custom bottom right border shall be used. [*Example*: Consider the following WordprocessingML markup for a custom bottom right border in a document: `<w:bottom w:val="custom" r:bottomRight="rIdCustomBottomRightBorder" …/>` The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rIdCustomBottomRightBorder` must contain the custom bottom right border image for the document. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).
id (Custom Defined Border Relationship Reference) Namespace: …/officeDocument/2006/relationships	Specifies the relationship ID for the relationship that contains the custom border image for the parent element. This custom border image is contained in a separate part within the WordprocessingML package. The relationship explicitly targeted by this attribute shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/image or the document shall be considered non-conformant. If this attribute is omitted, then no custom border shall be used. [*Example*: Consider the following WordprocessingML markup for a custom bottom border in a document: `<w:bottom w:val="custom" r:id="rIdCustomBottomBorder" …/>` The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rIdCustomBottomBorder` must contain the custom bottom border image for the document. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.7 Changed attribute for left element (Part 1, §17.6.7)

Attributes	Description
id (Custom Defined	Specifies the relationship ID for the relationship that contains the custom border image

Attributes	Description
Border Relationship Reference) Namespace: .../officeDocument/2006/relationships	for the parent element. This custom border image is contained in a separate part within the WordprocessingML package. The relationship explicitly targeted by this attribute shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/image or the document shall be considered non-conformant. If this attribute is omitted, then no custom border shall be used. [*Example*: Consider the following WordprocessingML markup for a custom bottom border in a document: `<w:bottom w:val="custom" r:id="rIdCustomBottomBorder" …/>` The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rIdCustomBottomBorder` must contain the custom bottom border image for the document. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.8 Changed attribute for printerSettings element (Part 1, §17.6.14)

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: http://schemas.openxmlformats.org/officeDocument/2006/customXml for the contentPart element http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer for the footerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/header for the headerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` contains

Attributes	Description
	the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.9 Changed attribute for right element (Part 1, §17.6.15)

Attributes	Description
id (Custom Defined Border Relationship Reference) Namespace: …/officeDocument/2006/relationships	Specifies the relationship ID for the relationship that contains the custom border image for the parent element. This custom border image is contained in a separate part within the WordprocessingML package. The relationship explicitly targeted by this attribute shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/image or the document shall be considered non-conformant. If this attribute is omitted, then no custom border shall be used. [*Example*: Consider the following WordprocessingML markup for a custom bottom border in a document: `<w:bottom w:val="custom" r:id="rIdCustomBottomBorder" …/>` The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rIdCustomBottomBorder` must contain the custom bottom border image for the document. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.10 Changed attribute for top element (Part 1, §17.6.21)

Attributes	Description
id (Custom Defined Border Relationship Reference) Namespace: …/officeDocument/2006/relationships	Specifies the relationship ID for the relationship that contains the custom border image for the parent element. This custom border image is contained in a separate part within the WordprocessingML package. The relationship explicitly targeted by this attribute shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/image or the document shall be considered non-conformant. If this attribute is omitted, then no custom border shall be used. [*Example*: Consider the following WordprocessingML markup for a custom bottom border in a document:

Attributes	Description
	`<w:bottom w:val="custom" r:id="rIdCustomBottomBorder" …/>` The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rIdCustomBottomBorder` must contain the custom bottom border image for the document. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).
topLeft (Custom Defined Top Left Border Relationship Reference) Namespace: …/officeDocument/2006/relationships	Specifies the relationship ID for the relationship that contains the custom top left border image for the parent element. This custom border image is contained in a separate part within the WordprocessingML package. The relationship explicitly targeted by this attribute shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/image or the document shall be considered non-conformant. If this attribute is omitted, then no custom top left border shall be used. [*Example*: Consider the following WordprocessingML markup for a custom top left border in a document: `<w:top w:val="custom" r:topLeft="rIdCustomTopLeftBorder" …/>` The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rIdCustomTopLeftBorder` must contain the custom top left border image for the document. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).
topRight (Custom Defined Top Right Border Relationship Reference) Namespace: …/officeDocument/2006/relationships	Specifies the relationship ID for the relationship that contains the custom top right border image for the parent element. This custom border image is contained in a separate part within the WordprocessingML package. The relationship explicitly targeted by this attribute shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/image or the document shall be considered non-conformant. If this attribute is omitted, then no custom top right border shall be used when the parent element is instantiated. [*Example*: Consider the following WordprocessingML markup for a custom top right border in a document: `<w:top w:val="custom" r:topRight="rIdCustomTopRightBorder" … />`

Attributes	Description
	The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rIdCustomTopRightBorder` must contain the custom top right border image for the document. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.11 Changed attribute for embedBold element (Part 1, §17.8.3.3)

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: http://schemas.openxmlformats.org/officeDocument/2006/customXml for the contentPart element http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer for the footerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/header for the headerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.12 Changed attribute for embedBoldItalic element (Part 1, §17.8.3.4)

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationshi	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: http://schemas.openxmlformats.org/officeDocument/2006/customXml for the contentPart element

Attributes	Description
ps	http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer for the footerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/header for the headerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.13 Changed attribute for embedItalic element (Part 1, §17.8.3.5)

Attributes	Description
id (Relationship to Part) Namespace: …/officeDocument/2006/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: http://schemas.openxmlformats.org/officeDocument/2006/customXml for the contentPart element http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer for the footerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/header for the headerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*]

Attributes	Description
	The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.14 Changed attribute for embedRegular element (Part 1, §17.8.3.6)

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: http://schemas.openxmlformats.org/officeDocument/2006/customXml for the contentPart element http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer for the footerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/header for the headerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.15 Changed attribute for footerReference element (Part 1, §17.10.2)

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: http://schemas.openxmlformats.org/officeDocument/2006/customXml for the contentPart element http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer for the footerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/heade

Attributes	Description
	r for the headerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.16 Changed attribute for headerReference element (Part 1, §17.10.5)

Attributes	Description
id (Relationship to Part) Namespace: …/officeDocument/2006/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: http://schemas.openxmlformats.org/officeDocument/2006/customXml for the contentPart element http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer for the footerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/header for the headerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type

Attributes	Description
	(Part 1, §22.8.2.1).

14.11.17 Changed attribute for dataSource element (Part 1, §17.14.9)

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: http://schemas.openxmlformats.org/officeDocument/2006/customXml for the contentPart element http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer for the footerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/header for the headerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.18 Changed attribute for headerSource element (Part 1, §17.14.16)

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: http://schemas.openxmlformats.org/officeDocument/2006/customXml for the contentPart element http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer for the footerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/header for the headerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/font

Attributes	Description
	for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.19 Changed attribute for recipientData element (Part 1, §17.14.28)

Attributes	Description
id (Relationship to Part) Namespace: …/officeDocument/2006/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: http://schemas.openxmlformats.org/officeDocument/2006/customXml for the contentPart element http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer for the footerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/header for the headerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.20 Changed attribute for src element (Part 1, §17.14.30)

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: http://schemas.openxmlformats.org/officeDocument/2006/customXml for the contentPart element http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer for the footerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/header for the headerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.21 Changed attribute for attachedTemplate element (Part 1, §17.15.1.6)

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: http://schemas.openxmlformats.org/officeDocument/2006/customXml for the contentPart element http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer for the footerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/header for the headerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyper

Attributes	Description
	link for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.22 Changed attribute for saveThroughXslt element (Part 1, §17.15.1.76)

Attributes	Description
id (XSL Transformation Location) Namespace: …/officeDocument/2006/relationships	Specifies an explicit relationship to the location of the XSL Transformation which shall be applied. The relationship targeted by this element shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/transform, or this document shall be declared non-conformant. [*Example*: Consider a XML document that must have the XSL transform located at `c:\Example Transform.xslt` applied when the document is saved as a single XML file. This requirement would be specified using the following WordprocessingML in the document settings: `<w:saveThroughXslt r:id="rId5" />` The saveThroughXslt element specifies that the relationship located at `rId5` must be used when saving as a single XML file in this case, that relationship must target `c:\Example Transform.xslt`. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.23 Changed attribute for longDesc element (Part 1, §17.15.2.23)

Attributes	Description
id (Relationship to Part) Namespace: …/officeDocument/2006/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: http://schemas.openxmlformats.org/officeDocument/2006/customXml for the contentPart element http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer

Attributes	Description
	for the footerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/header for the headerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.24 Changed attribute for sourceFileName element (Part 1, §17.15.2.39)

Attributes	Description
id (Relationship to Part) Namespace: …/officeDocument/2006/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: http://schemas.openxmlformats.org/officeDocument/2006/customXml for the contentPart element http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer for the footerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/header for the headerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*]

Attributes	Description
	The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.25 Changed attribute for subDoc element (Part 1, §17.17.1.1)

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: http://schemas.openxmlformats.org/officeDocument/2006/customXml for the contentPart element http://schemas.openxmlformats.org/officeDocument/2006/relationships/ for the footerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/header for the headerReference element http://schemas.openxmlformats.org/officeDocument/2006/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

14.11.26 Changed attribute for altChunk element (Part 1, §17.17.2.1)

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID to a specified part containing alternate content for import. If the specified relationship does not match the relationship type required by the parent element, then this document shall be considered to be non-conformant. [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />`

Attributes	Description
	The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§Part 1, §22.8.2.1).

15. SpreadsheetML Reference Material

[*Note*: For further information on the mapping of elements and attributes to OPC parts, see the Bibliography entry, "Information on elements, attributes, and OPC parts in ISO/IEC 29500 (OOXML)". *end note*]

15.1 Table of Contents

This subclause is informative.

End of informative text.

15.2 Workbook

15.2.1 Additional attribute for fileSharing element (Part 1, §18.2.12)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description
reservationPassword (Write Reservation Password)	Specifies the legacy hash of the password required for editing this workbook. `The hash is generated using the logic defined in the` revisionsPassword `attribute of the` wookbookProtection `element (Part 1, §18.2.29).` The possible values for this attribute are defined by the ST_UnsignedShortHex simple type (§15.7.2).

15.2.2 Additional attribute for webPublishing element (Part 1, §18.2.24)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description
codePage (Code Page)	This attribute is used only for compatibility with the existing corpus of binary documents, and is ignored if the characterSet attribute is present. Specifies the encoding the application uses when a Web page is saved. A code page is a table that relates the binary character codes used by a program to keys on the keyboard or to the appearance of characters on the display. Code pages are a means of providing support for the languages used in different countries. [*Note*: There are a number of code page technologies. One example of potential values can be found at: http://www.unicode.org/Public/MAPPINGS/ *end note*] The default value for this attribute is the workbook's encoding. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

15.2.3 Additional attributes for workbookProtection element (Part 1, §18.2.29)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description
revisionsPassword (Legacy Revisions Password)	Specifies the legacy hash of the password required for unlocking revisions in this workbook. The hash is generated from an 8-bit wide character. The input string shall be in UTF-16LE format (if there is a leading BOM character (U+FEFF) in the encoded password it is removed before hash calculation), and these 16-bit Unicode characters shall be converted down to 8 bits before the hash is computed, using the following logic: [*Note*: This legacy conversion attempts to fit UTF-16 encoded characters into a single-

Attributes	Description

	byte character set. As such, if the input string uses characters from multiple character sets, many characters are unmapped in the destination character set and take on the default value, 0x3F. For this reason, it is recommended that applications choose a character set which maps the maximum number of characters from the input string and explicitly declare the character set used in the revisionsCharacterSet attribute. Not doing so will inhibit interoperability. *end note*]
	
	For SpreadsheetML password hash purposes, Unicode UTF-16 input code points are converted to a single or double byte character set.
	
	Code points with no representation in the target character set are replaced with Unicode character 0x3f (?).
	
	The values permitted by this attribute are names and aliases listed in the IANA character set listing found at http://www.iana.org/assignments/character-sets.
	For single byte character sets, each Unicode code point is replaced by a single byte or 0x3f if an appropriate character doesn't exist in the character set.
	
	For double byte character sets, each Unicode code point is replaced by either a single byte, or a two byte sequence, depending on the input character, or 0x3f if an appropriate character doesn't exist in the character set. In our tables the target is a single byte sequence if the most significant byte is 0x00, otherwise it is a double byte sequence, with the lead byte being the most significant byte.
	
	To convert, first check if conversion is being done to a single or double byte code page and load the appropriate WCTABLE code page table.
	
	For each input character, look up the code point in the WCTABLE. There are 3 possibilities: Not found, single byte, or double byte.
	
	• If the input character is not found, append 0x3f and continue to the next character.
	• If the result is a single byte, check to make sure the entry in the MBTABLE matches the input. If it matches, append the single byte to the output. If it does not match, append 0x3f to the output.
	• If the result is a double byte,check to make sure the entry in the DBCSENTRY table for the appropriate lead byte matches the input character. If it matches, append the lead byte and trail byte to the output. If it does not match, append 0x3f to the output.
	
	The following pseudocode describes how this conversion should be done:
	
	`int WideCharToMultiByte(wchar_t* wszInput, byte* szOutput)`
	`{`
	`  // Remember output start so we can return length`
	`  byte* szOutputStart = szOutput;`

Attributes	Description

```
    // Load Character Set Tables and determine
    // double/single byte nature.
    // This will depend on how the character sets are
represented on
    // the target machine.  TABLECLASS represents some abstract
    // representation of this structure here.
    TABLECLASS pTables = LoadCharacterSetTables();
    Bool bDoubleByte = IsCharacterSetDoubleByte();


    while (*wszInput != 0)
    {
        if (bDoubleByte)
            szOutput = AppendDoubleByte(pTables, *wszInput,
szOutput);
        else
            szOutput = AppendSingleByte(pTables, *wszInput,
szOutput);

        // Read next input wchar_t
        wszInput++;
    }

    // Null terminate the output
    *szOutput = 0;

    // Return output length
    return szOutput – szOutputStart;
}

byte* AppendSingleByte(TABLECLASS pTables, wchar_t wcIn, byte*
szOutput)
{
    // Look up byte that we want to append.
    byte bOut = pTables->LookUpSingleByte(wcIn);

    // Make sure that bOut matches the input, otherwise use ?
    // (ie: no best fit behavior allowed)
    if (wcIn != pTables->LookUpWideChar(bOut))
        bOut = 0x3f;

    *szOutput = bOut;
    szOutput++;
    return szOutput;
}
```


Attributes	Description

	`byte* AppendDoubleByte(TABLECLASS pTables, wchar_t wcIn, byte* szOutput)`
	`{`
	`  // Look up bytes that we want to append.`
	`  UINT16 bytesOut = pTables->LookUpDoubleByte(wcIn);`
	
	`  // See if it is a single or double byte sequence`
	`  if (bytesOut & 0xFF00)`
	`  {`
	`  // It is a double byte sequence`
	`  // Make sure that bytesOut matches the input, otherwise use ?`
	`  // (ie: no best fit behavior allowed)`
	`  if (wcIn != pTables->LookUpWideChar(bytesOut))`
	`  {`
	`  // Use ?, it will be added below`
	`  bytesOut = 0x003f;`
	`  }`
	`  else`
	`  {`
	`  // It matched, use the lead byte we found`
	`  // trail byte will be added below`
	`  *szOutput = bytesOut >> 8;`
	`  szOutput++;`
	`  }`
	`  else`
	`  {`
	`  // It is a single byte sequence`
	`  // Make sure that bytesOut matches the input, otherwise use ?`
	`  // (ie: no best fit behavior allowed)`
	`  if (wcIn != pTables->LookUpWideChar(bytesOut & 0xFF))`
	`  bytesOut = 0x003f;`
	`  }`
	
	`  // Add the single or trail byte`
	`  *szOutput = bytesOut & 0xFF;`
	`  szOutput++;`
	
	`  return szOutput;`
	`}`
	
	`class pTables`
	`{`
	`  // Construction depends on how you choose to store & load the`
	`  // table files`

Attributes	Description

	`byte LookUpSingleByte(wchar_t wcIn)`
	`{`
	`// How you access the table depends on your storage mechanism.`
	`// Look up the line in WCTABLE where the first column matches wcIn,`
	`// and then return the byte value from the second column.`
	`if (exists WCTABLE{wcIn})`
	`return WCTABLE{wcIn}.SecondColumn;`
	
	`// If it doesn’t exist, return ?`
	`return 0x3f;`
	`}`
	
	`UINT16 LookUpDoubleByte(wchar_t wcIn)`
	`{`
	`// How you access the table depends on your storage mechanism.`
	`// Look up the line in WCTABLE where the first column matches wcIn,`
	`// and then return the double byte value from the second column.`
	`if (exists WCTABLE{wcIn})`
	`return WCTABLE{wcIn}.SecondColumn;`
	
	`// If it doesn’t exist, return ?`
	`return 0x003f;`
	`}`
	
	`// Overload that looks up wide chars from single byte code points.`
	`wchar_t LookUpWideChar(byte bIn)`
	`{`
	`// How you access the table depends on your storage mechanism.`
	`// Look up the line in MBTABLE where the first column matches bIn,`
	`// and then return the wchar_t value from the second column.`
	`if (exists MBTABLE{bIn})`
	`return MBTABLE{bIn}.SecondColumn;`
	
	`// If it doesn’t exist, return ?`
	`return 0x003f;`
	`}`
	
	`// Overload that looks up wide chars from double byte code`

Attributes	Description

	points
	wchar_t LookUpWideChar(UINT16 bytesIn)
	{
	// How you access the table depends on your storage mechanism.
	// First find the DBCSTABLE where the LeadByte matches
	// the lead (most significant) input byte.
	if (exists DBCSTABLE{bytesIn >> 8))
	{
	DbcsTable = DBCSTABLE{bytesIn >> 8);
	
	// Look up the line in DbcsTable where the first column
	// matches the input trail (least significant) byte,
	// and then return the wchar_t value from the second column.
	if (exists DbcsTable{bytesIn & 0xFF})
	return DbcsTable{bytesIn & 0xFF}.SecondColumn;
	}
	
	// Either the lead byte table or specific trail byte
	// doesn't exist in the table, return ?
	return 0x003f;
	}
	}
	
	The resulting value is hashed using the low-order word algorithm defined in §14.7.1. This step assumes that all words are unsigned, the word size is two bytes, and that bit-level shift-left/shift-right operations shift in the direction of the highest-order and lowest-order bit, respectively. [*Example*: `0x61 SHR 1` is `0xC2`, as `01100001` shifted one position in the direction of its highest-order bit is `11000010`. *end example*]
	
	[*Example*: This algorithm can be represented by the following pseudocode:
	
	// Function Input:
	// szPassword: NULL terminated C-Style string
	// cchPassword: The number of characters in szPassword (not including the NULL terminator)
	unsigned_short GetPasswordHash(const char *szPassword, int cchPassword) {
	unsigned_short wPasswordHash;
	const char *pch;
	
	wPasswordHash = 0;
	
	if (cchPassword > 0)
	{

Attributes	Description

	`pch = &szPassword[cchPassword];`
	`while (pch-- != szPassword)`
	`{`
	`wPasswordHash = ((wPasswordHash >> 14) & 0x01) | ((wPasswordHash << 1) & 0x7fff);`
	`wPasswordHash ^= *pch;`
	`}`
	`wPasswordHash = ((wPasswordHash >> 14) & 0x01) | ((wPasswordHash << 1) & 0x7fff);`
	`wPasswordHash ^= cchPassword;`
	`wPasswordHash ^= (0x8000 | ('N' << 8) | 'K');`
	`}`
	
	`return(wPasswordHash);`
	`}`
	end example]
	
	The possible values for this attribute are defined by the ST_UnsignedShortHex simple type (§15.7.2).

revisionsPassword CharacterSet (Revisions Password Character Set)	Name of the character set associated with the legacy revisionsPassword hash. The values permitted by this attribute are names and aliases listed in the IANA CHARACTER SETS listing found at http://www.iana.org/assignments/character-sets. The possible values for this attribute are defined by the W3C XML Schema string datatype.
workbookPassword (Legacy Workbook Password)	Specifies the legacy hash of the password required for unlocking revisions in this workbook. The hash is generated using the logic defined in the preceding revisionsPassword attribute. The possible values for this attribute are defined by the ST_UnsignedShortHex simple type (§15.7.2).
workbookPasswordCharacterSet (Workbook Password Character Set)	Name of the character set associated with the workbookPassword hash. The values permitted by this attribute are the names and aliases listed in the IANA CHARACTER SETS listing found at http://www.iana.org/assignments/character-sets. The possible values for this attribute are defined by the W3C XML Schema string datatype.

15.2.4 Modified content for Date Conversion for Serial Date-Times (Part 1, §18.17.4.1)

When interpreting a document of a transitional conformance class, Part 1, §18.17.4.1 is replaced by the following text:

A *serial date-time* is a number that represents a date and time. This signed value is in units of days relative to the base date for the selected date system. Serial date-times increase by 1 into each successive day, and decrease by 1 into each preceding day. Fractional portions of serial date-times represent fractions of a single day. [*Example*: When using the 1900 date system, which has a base date of 30th December 1899, a serial date-time of 1.5 represents midday on the 31st December 1899 (serial date-time day 1); that is, 1899-12-31T12:00. A serial date-time of -4.25 represents 6 pm on the 25th December 1899; that is, 1899-12-25T18:00. *end example*] The base dates and the related serial date-times represent local date and time.

Two different bases are used for converting dates to and from serial date-times:

- In the *1900 date system*, the lower limit is January 1, 1900, 00:00:00, which has a serial date-time of 1. The upper limit is December 31, 9999, 23:59:59, which has a serial date-time of 2,958,465.9999884. The base date for this date base system is December 31, 1899, which has a serial date-time of 0.
- In the *1904 date system*, the lower limit is January 1st, 0001, 00:00:00, which has a serial date-time of -695055. The upper limit is December 31st, 9999, 23:59:59.999, which has a serial date-time of 2,957,003.9999884. The base date for this system is midnight (00:00:00) on the morning of January 1st, 1904, which has a serial date-time of 0.

A serial date-time outside the temporal range for the selected date system is invalid.

The date system is specified by the value of the date1904 attribute of the workbookPr element. [*Example*:

1900 date system: `<workbookPr showObjects="all"/>`
1904 date system: `<workbookPr date1904="1" showObjects="all"/>`

end example]

15.3 Worksheets

15.3.1 Worksheets

15.3.1.1 legacyDrawing (Legacy Drawing Reference)

This element is present when the sheet contains drawing shapes defined by VML. In this case, the element contains an explicit relationship whose ID points to the part containing the VML definitions.

[*Example*:

```
<drawing r:id="rId1"/>
```

end example]

Attributes	Description
id (Relationship Id) Namespace: …/officeDocument	This value references a relationship Id for the sheet. The relationship shall point to the part containing the VML definition. The possible values for this attribute are defined by the ST_RelationshipId simple type

Attributes	Description
/2006/relationships	(Part 1, §22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LegacyDrawing) is located in §A.2. *end note*]

15.3.1.2 legacyDrawingHF (Legacy Drawing Reference in Header Footer)

This element specifies the explicit relationship to the part containing the VML defining pictures rendered in the header / footer of the sheet.

Attributes	Description
id (Relationship Id) Namespace: .../officeDocument/2006/relationships	This value references a relationship Id for the sheet. The relationship shall point to the part containing the VML definition. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LegacyDrawing) is located in §A.2. *end note*]

15.3.1.3 Additional attribute for dataConsolidate element (Part 1, §18.3.1.29)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description
leftLabels (Starting Column Labels)	Semanticaly equivlent to startLabels. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

15.3.1.4 Additional attributes for protectedRange element (Part 1, §18.3.1.71)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description
password (Legacy Password)	Specifies the legacy hash of the password required for editing this range. The hash is generated using the logic defined in the revisionsPassword attribute of the workbookProtection element (Part 1, §18.2.29). The possible values for this attribute are defined by the ST_UnsignedShortHex simple type (§15.7.2).

Attributes	Description

securityDescriptor (Security Descriptor)	Optional setting to specify the relative security descriptor. The security descriptor defines user accounts who can edit this range without providing a password to access the range.
	
	The format of a securityDescriptor is application defined; however, it is recommended that the following format be used for interoperability between implementations:
	• username@domain
	
	If multiple user accounts are specified in the securityDescriptor attribute, each account shall be delimited by parentheses.
	
	[*Example*: This example demonstrates two user accounts in the security descriptor attribute:
	
	`<protectedRanges>`
	`  <protectedRange sqref="A1:C5" name="Range1"`
	`securityDescriptor="(user1@iso.org)(user2@iso.org)"/>`
	`</protectedRanges>`
	
	end example]
	
	If an application is unable to resolve the meaning of the securityDescriptor, it shall treat the attribute as if it had been removed.
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

15.3.1.5 Additional attribute for sheetProtection element (Part 1, §18.3.1.84)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description
password (Password)	Specifies the hash of the password required for editing this chart sheet. `The hash is generated using the logic defined in the` revisionPassword `attribute of the` workbookProtection `element (Part 1, §18.2.29).` The possible values for this attribute are defined by the ST_UnsignedShortHex simple type (§15.7.2).

15.3.1.6 Additional attribute for sheetProtection element (Part 1, §18.3.1.85)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description
password (Legacy Password)	Specifies the legacy hash of the password required for editing this worksheet.

Attributes	Description
	The hash is generated using the logic defined in the revisionsPassword attribute of the workbookProtection element (Part 1, §18.2.29). The possible values for this attribute are defined by the ST_UnsignedShortHex simple type (§15.7.2).

15.3.2 AutoFilter Settings

15.3.2.1 Attributes with modified descriptions for dynamicFilter element (Part 1, §18.3.2.5)

The following attributes have modified descriptions when specified for a document of a transitional conformance class:

Attributes	Description
maxVal (Max Value)	A maximum value for dynamic filter. maxVal/maxValIso shall be required for today, yesterday, tomorrow, nextWeek, thisWeek, lastWeek, nextMonth, thisMonth, lastMonth, nextQuarter, thisQuarter, lastQuarter, nextYear, thisYear, lastYear, and yearToDate. The above criteria are based on a value range. [*Example*: If today's date is September 22nd, then the range for thisWeek is the values greater than or equal to September 17 and less than September 24. *end example*] In the thisWeek range, the lower value is expressed using val or valIso. The higher value is expressed using maxVal or maxValIso. These dynamic filter shall not require val/valIso or maxVal/maxValIso: Q1, Q2, Q3, Q4, M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11 and M12. The above criteria shall not specify the range using val/valIso and maxVal/maxValIso because Q1 always starts from M1 to M3, and M1 is always January. These types of dynamic filters shall use val and shall not use maxVal/maxValIso: aboveAverage and belowAverage. If maxValIso and maxVal are both present, maxValIso shall take precedence. The possible values for this attribute are defined by the W3C XML Schema double datatype.
val (Value)	A minimum numeric or serial date value for dynamic filter. (See description of ValIso to understand when val is required.) If valIso and val are both present, valIso shall take precedence. The possible values for this attribute are defined by the W3C XML Schema double datatype.
valIso (ISO Value)	A minimum date value for dynamic filter. (See description of maxVal/maxValIso to

Attributes	Description
	understand when val/valIso is required.) The possible values for this attribute are defined by the W3C XML Schema dateTime datatype.

15.4 Styles

15.4.1 left (Leading Edge Border)

Semantically equivalent to start (Part 1, §18.8.37).

Attributes	Description
style (Line Style)	The line style for this border. The possible values for this attribute are defined by the ST_BorderStyle simple type (Part 1, §18.18.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BorderPr) is located in §A.2. *end note*]

15.4.2 right (Trailing Edge Border)

Semantically equivalent to end (Part 1, §18.8.16).

Attributes	Description
style (Line Style)	The line style for this border. The possible values for this attribute are defined by the ST_BorderStyle simple type (Part 1, §18.18.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BorderPr) is located in §A.2. *end note*]

15.5 Pivot Tables

15.5.1 Pivot Tables

15.5.1.1 Additional attribute for pivotCacheDefinition element (Part 1, §18.10.1.67)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description
refreshedDate (PivotCache Last	Specifies the date when the cache was last refreshed. This attribute depends on whether the application exposes mechanisms via the user interface whereby the end-user can

Attributes	Description
Refreshed Date)	refresh the cache. If refreshedDateIso and refreshedDate are both present, refreshedDateIso shall take precedence. The possible values for this attribute are defined by the W3C XML Schema double datatype.

15.6 External Data Connections

15.6.1 Additional attribute for textPr element (Part 1, §18.13.12)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description
codePage (Code Page)	Code page associated with the text file. This attribute is used only for backwards compatibility, and is ignored if the characterSet attribute is present. [*Note*: There are a number of code page technologies. One example of potential values can be found at: http://www.unicode.org/Public/MAPPINGS *end note*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

15.7 Simple Types

The following additional simple type information in the http://schemas.openxmlformats.org/spreadsheetml/2006/main namespace is used for documents of a transitional conformance class.

15.7.1 Additional enumeration values for ST_PivotAreaType (Part 1, §18.18.58)

The following addtional enumeration values can be specified for a document of a transitional conformance class.

Enumeration Value	Description
topRight (Top Corner, Trailing Edge)	Semantically equivalent to topEnd.

15.7.2 ST_UnsignedShortHex (Unsigned Short Hex)

This simple type defines the Hex representation of an unsigned short.

This simple type's contents are a restriction of the W3C XML Schema hexBinary datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a length of exactly 4 hexadecimal digit(s).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_UnsignedShortHex) is located in §A.2. *end note*]

15.7.3 Removed enumeration values for ST_CellType (Part 1, §18.18.11)

For transitional documents, the restriction on the simple type ST_CellType having the value "d" (ISO 8601 format) is removed.

15.8 Formulas

15.8.1 Attribute synonym for c element (Part 1, §18.6.1)

The following additional attribute can be specified for a document of a transitional conformance class:

Attributes	Description
ref (Cell Reference)	An A-1 style reference to a cell. The possible values for this attribute are defined by the ST_CellRef simple type (Part 1, §18.18.7).

This attribute is semantically equivalent to r (Part 1, §18.6.1).

Only one or the other of r and ref can be defined in any given instance.

15.8.2 Additional representation for dates and times (Part 1, §18.17.4)

For a document of a transitional conformance class, each unique instant in SpreadsheetML time shall be stored as an ISO 8601-formatted string or as a serial value.

15.9 Changed attributes

The following attributes, which are defined in subclauses within Part 1, §18, "SpreadsheetML", have different source relationships when used in documents of the Transitional conformance class:

15.9.1 Changed attribute for externalReference element (Part 1, §18.2.8)

Attributes	Description
id (Relationship Id) Namespace: …/officeDocument/2006/relationships	Specifies a unique identifier that is used to identify a relationship to another part in the file. Relationship identifiers link the element definition with the part where data for the element is stored. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.2 Changed attribute for pivotCache element (Part 1, §18.2.17)

Attributes	Description
id (Relationship Id) Namespace: …/officeDocument	Specifies the identifier to a pivot cache definition part where cached data is stored. This attribute is required.

Attributes	Description
/2006/relationships	The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.3 Changed attribute for sheet element (Part 1, §18.2.19)

Attributes	Description
id (Relationship Id) Namespace: .../officeDocument/2006/relationships	Specifies the identifier of the sheet part where the definition for this sheet is stored. This attribute is required. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.4 Changed attribute for control element (Part 1, §18.3.1.19)

Attributes	Description
id (Relationship Id) Namespace: .../officeDocument/2006/relationships	This relationship ID references an Embedded Control Data part that contains control-specific properties and state information about this particular embedded control. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.5 Changed attribute for controlPr element (Part 1, §18.3.1.20)

Attributes	Description
id (Relationship ID for Embedded Control Properties) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID for the relationship which contains the properties for this embedded control. This property bag is contained in a separate part within the package. The relationship explicitly targeted by this attribute shall be of relationship type http://schemas.openxmlformats.org/officeDocument/2006/relationships/control or the document shall be considered non-conformant. If this attribute is omitted, then the embedded control shall be given no property bag when instantiated. [*Example*: Consider the following WordprocessingML markup for an embedded control in a document: `<w:control r:id="rId5" w:id="CheckBox1" w:name="CheckBox1" w:shapeid="_x0000_s1027" w:class="shape" w:w="145" w:h="28" w:align="left" />` The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rId5` must contain the property data for this embedded control. *end example*]

Attributes	Description
	The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.6 Changed attribute for customPr element (Part 1, §18.3.1.22)

Attributes	Description
id (Relationship Id) Namespace: .../officeDocument/2006/relationships	This relationship references the binary part containing the specified custom properties. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.7 Changed attribute for dataRef element (Part 1, §18.3.1.30)

Attributes	Description
id (relationship Id) Namespace: .../officeDocument/2006/relationships	Used only when the source range is external to this workbook. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.8 Changed attribute for drawing element (Part 1, §18.3.1.36)

Attributes	Description
id (Relationship id) Namespace: .../officeDocument/2006/relationships	Relationship Id referencing a part containing DrawingML definitions for this worksheet. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.9 Changed attribute for drawingHF element (Part 1, §18.3.1.37)

Attributes	Description
id (Relationship ID for Embedded Control Properties) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID for the relationship to the DrawingML part that contains the drawing objects used in the header and footer. This DrawingML part is a separate part within the package. [*Example*: `<drawingHF r:id="rId2" lho="7" lhf="6"/>` The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rId5` must contain the drawing objects used in the header and

Attributes	Description
	footer. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.10 Changed attribute for hyperlink element (Part 1, §18.3.1.47)

Attributes	Description
id (Relationship Id) Namespace: .../officeDocument/2006/relationships	Relationship Id in this sheet's relationships part, expressing the target location of the resource. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.11 Changed attribute for objectPr element (Part 1, §18.3.1.56)

Attributes	Description
id (Relationship ID to Embedded Object Data) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID for the relationship that targets the Embedded Object Part containing the embedded object data. The specified relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/oleObject or the document shall be considered non-conformant. [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` targets the part containing the corresponding embedded object information. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.12 Changed attribute for oleObject element (Part 1, §18.3.1.59)

Attributes	Description
id (Relationship Id) Namespace: .../officeDocument/2006/relationships	Relationship Id of the relationship pointing to the object persistence part. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.13 Changed attribute for pageSetup element (Part 1, §18.3.1.63)

Attributes	Description
id (Id) Namespace: .../officeDocument/2006/relationships	Relationship Id of the devMode printer settings part. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.14 Changed attribute for pageSetup element (Part 1, §18.3.1.64)

Attributes	Description
id (Id) Namespace: .../officeDocument/2006/relationships	Relationship Id of the devMode printer settings part. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.15 Changed attribute for picture element (Part 1, §18.3.1.67)

Attributes	Description
id (Relationship Id) Namespace: .../officeDocument/2006/relationships	Relationship Id pointing to the image part. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.16 Changed attribute for pivotSelection element (Part 1, §18.3.1.69)

Attributes	Description
id (Relationship Id) Namespace: .../officeDocument/2006/relationships	Relationship Id pointing to the particular PivotTable Part corresponding to this selection. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.17 Changed attribute for tablePart element (Part 1, §18.3.1.94)

Attributes	Description
id (Relationship Id) Namespace: .../officeDocument/2006/relationshi	This relationship Id is used to locate a particular table definition part. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

Attributes	Description
ps	

15.9.18 Changed attribute for pivotCacheDefinition element (Part 1, §18.10.1.67)

Attributes	Description
id (Relationship Identifier) Namespace: .../officeDocument/2006/relationships	Specifies the unique identifier that corresponds to the related `pivotCacheRecords` part. See (Part 1, §18.10.1.68) for more information. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.19 Changed attribute for rangeSet element (Part 1, §18.10.1.79)

Attributes	Description
id (Relationship Id) Namespace: .../officeDocument/2006/relationships	Specifies the unique identifier of the Workbook part where the range set is stored. See Workbook (Part 1, §18.2) for more information. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.20 Changed attribute for worksheetSource element (Part 1, §18.10.1.95)

Attributes	Description
id (Relationship Id) Namespace: .../officeDocument/2006/relationships	Specifies the identifier to the Sheet part whose data is stored in the cache. See the Sheet section (Part 1, §18.2) for more information. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.21 Changed attribute for header element (Part 1, §18.11.1.1)

Attributes	Description
id (Relationship ID) Namespace: .../officeDocument/2006/relationships	This is the ID that is used to find the corresponding log record of the changes made for this header. Use the corresponding relationship expressed in the revisionHeaders part to locate the log record that lists the specific changes. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.22 Changed attribute for externalBook element (Part 1, §18.14.7)

Attributes	Description
id (Relationship to supporting book file path) Namespace: .../officeDocument/2006/relationships	Relationship ID that references a link in the relationships collection. The target attribute in the associated relationship will specify the worksheet XML file in the current SpreadsheetML document ZIP archive that makes use of this externalbook. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

15.9.23 Changed attribute for oleLink element (Part 1, §18.14.11)

Attributes	Description
id (Object Link Relationship) Namespace: .../officeDocument/2006/relationships	Relationship ID that references a link in the relationships collection. The target attribute in the associated relationship will specify the external file name used for this oleLink. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

16. PresentationML Reference Material

[*Note*: For further information on the mapping of elements and attributes to OPC parts, see the Bibliography entry, "Information on elements, attributes, and OPC parts in ISO/IEC 29500 (OOXML)". *end note*]

16.1 Table of Contents

This subclause is informative.

End of informative text.

16.2 Presentation

16.2.1 Presentation Properties

16.2.1.1 htmlPubPr (HTML Publishing Properties)

This element specifies the publishing properties to be used when publishing this presentation document to the HTML file format. The target output profile is identified by the contents of the target attribute.

Attributes	Description
id (Publish Path) Namespace: .../officeDocument/2006/relationships	Specifies the path that should be used when publishing. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).
showSpeakerNotes (Show Speaker Notes)	Specifies whether to show speaker notes when publishing. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
target (Target Output Profile)	Specifies the version of HTML output targeted by the output of any web page produced by this document. This attribute shall only contain a string that represents an output profile defined by published standards and W3C recommendations. Product names shall not be used to define a profile. The following reserved values and their targets are listed below: **Value** | **Target**

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>W3C XHTML+CSS1</td><td>W3C XHTML 1.0 + CSS 1</td></tr>
<tr><td>W3C HTML4+CSS1</td><td>W3C HTML 4.01 + CSS 1</td></tr>
<tr><td>W3C XHTML+CSS2</td><td>W3C XHTML 1.0 + CSS 2</td></tr>
<tr><td>W3C HTML4+CSS2</td><td>W3C HTML 4.01 + CSS 2</td></tr>
</table>

[Example: For example, consider the following set of HTML publishing settings:
<pre>
<p:htmlPubPr … target="W3C HTML4+CSS2">
  …
</p:htmlPubPr>
</pre>
The target attribute explicitly declares that any web page generated from this document should target the W3C HTML4+CSS2 profile. end example]

The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>title (HTML Output Title)</td><td>Specifies a title for the HTML output file.

The possible values for this attribute are defined by the W3C XML Schema string datatype.</td></tr>
</table>

[*Note*: The W3C XML Schema definition of this element's content model (CT_HtmlPublishProperties) is located in §A.3. *end note*]

16.2.1.2 webPr (Web Properties)

This element specifies all general output properties that pertain to generating a web format version of the presentation document.

Attributes	Description
allowPng (Allow PNG in HTML output)	Specifies whether to allow the output of PNG format pictures in the HTML document. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
clr (Slide Navigation Colors for HTML output)	Specifies the color constraints that are to be used when generating HTML output. The possible values for this attribute are defined by the ST_WebColorType simple type (§16.4.1).
encoding (Encoding for HTML output)	Specifies the particular HTML character set encoding that should be used when generating output.

Attributes	Description
	The possible values for this attribute are defined by the ST_WebEncoding simple type (§16.4.2).
imgSz (Image size for HTML output)	Specifies the screen size for which the images in the HTML output should be optimized. The possible values for this attribute are defined by the ST_WebScreenSize simple type (§16.4.3).
organizeInFolders (Organize HTML output in folders)	Specifies whether the supporting output files should be automatically organized into a folder. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
relyOnVml (Rely on VML for HTML output)	Specifies whether graphics should be output in VML within the HTML. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
resizeGraphics (Resize graphics in HTML output)	Specifies whether to resize graphics to fit within the browser window when generating the HTML output. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showAnimation (Show animation in HTML output)	Specifies whether to show presentation animation in the HTML output file. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
useLongFilenames (Use long file names in HTML output)	Specifies whether to allow the use of long file names when generating the HTML output. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_WebProperties) is located in §A.3. *end note*]

16.2.1.3 Additional attributes for modifyVerifier element (Part 1, §19.2.1.19)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description
algIdExt (Cryptographic Algorithm Extensibility)	Specifies that a cryptographic algorithm which was not defined by ISO/IEC 29500 has been used to generate the hash value stored with this document. This value, when present, shall be interpreted based on the value of the algIdExtSource attribute in order to determine the algorithm used, which shall be application-defined. [*Rationale*: This extensibility affords the fact that with exponentially increasing

Attributes	Description

	computing power, documents created in the future might need to utilize as yet undefined hashing algorithms in order to remain secure. *end rationale*]
	
	If this value is present, the cryptAlgorithmClass, cryptAlgorithmType, and cryptAlgorithmSid attribute values shall be ignored in favor of the algorithm defined by this attribute.
	
	[*Example*: Consider a PresentationML document with the following information stored in its protection element:
	
	`<… p:algIdExt="0000000A"`
	`p:algIdExtSource="futureCryptography"`
	`p:hashData="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The algIdExt attribute value of `0000000A` specifies that the algorithm with hex code `A` shall be used as defined by the `futureCryptography` application. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

algIdExtSource (Algorithm Extensibility Source)	Specifies the application which defined the algorithm value specified by the algIdExt attribute.
	
	[*Example*: Consider a PresentationML document with the following information stored in one its protection element:
	
	`<… p:algIdExt="0000000A"`
	`p:algIdExtSource="futureCryptography"`
	`p:hashData="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The algIdExtSource attribute value of `futureCryptography` specifies that the algorithm used here was published by the `futureCryptography` application. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

cryptAlgorithmClass (Cryptographic Algorithm Class)	Specifies the class of cryptographic algorithm used by this protection. [*Note*: The initial version of ISO/IEC 29500 only supports a single version - `hash` - but future versions can expand this as necessary. *end note*]
	
	[*Example*: Consider a PresentationML document with the following information stored in its protection element:
	
	`<… p:cryptAlgorithmClass="hash"`
	`p:cryptAlgorithmType="typeAny"`
	`p:cryptAlgorithmSid="1"`
	`p:hashData="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>The cryptAlgorithmClass attribute value of hash specifies that the algorithm used for the password is a hashing algorithm. end example]

The possible values for this attribute are defined by the ST_AlgClass simple type (§20.1.2.1).</td></tr>
<tr><td>cryptAlgorithmSid (Cryptographic Hashing Algorithm)</td><td>Specifies the specific cryptographic hashing algorithm which shall be used along with the saltData attribute and user-supplied password in order to compute a hash value for comparison.

The possible values for this attribute shall be interpreted as follows:
<table>
<tr><th>Value</th><th>Algorithm</th></tr>
<tr><td>1</td><td>MD2</td></tr>
<tr><td>2</td><td>MD4</td></tr>
<tr><td>3</td><td>MD5</td></tr>
<tr><td>4</td><td>SHA-1</td></tr>
<tr><td>5</td><td>MAC</td></tr>
<tr><td>6</td><td>RIPEMD</td></tr>
<tr><td>7</td><td>RIPEMD-160</td></tr>
<tr><td>8</td><td>Undefined. Shall not be used.</td></tr>
<tr><td>9</td><td>HMAC</td></tr>
<tr><td>10</td><td>Undefined. Shall not be used.</td></tr>
<tr><td>11</td><td>Undefined. Shall not be used.</td></tr>
<tr><td>12</td><td>SHA-256</td></tr>
<tr><td>13</td><td>SHA-384</td></tr>
<tr><td>14</td><td>SHA-512</td></tr>
<tr><td>Any other value</td><td>Undefined. Shall not be used.</td></tr>
</table>
[Example: Consider a PresentationML document with the following information stored in its protection element:

<pre><… p:cryptAlgorithmClass="hash"
  p:cryptAlgorithmType="typeAny"
  p:cryptAlgorithmSid="1"
  p:hashData="9oN7nWkCAyEZib1RomSJTjmPpCY=" /></pre>
The cryptAlgorithmSid attribute value of 1 specifies that the SHA-1 hashing algorithm shall be used to generate a hash from the user-defined password. end example]

The possible values for this attribute are defined by the W3C XML Schema unsignedInt</td></tr>
</table>

Attributes	Description
	datatype.

cryptAlgorithmType (Cryptographic Algorithm Type)	Specifies the kind of cryptographic algorithm used by this protection. [*Note*: The initial version of ISO/IEC 29500 only supports a single type - `typeAny` - but future versions can expand this as necessary. *end note*]
	
	[*Example*: Consider a PresentationML document with the following information stored in its protection element:
	
	`<… p:cryptAlgorithmClass="hash"`
	`p:cryptAlgorithmType="typeAny"`
	`p:cryptAlgorithmSid="1"`
	`p:hashData="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The cryptAlgorithmType attribute value of `typeAny` specifies that any algorithm type might have been used for the password. *end example*]
	
	The possible values for this attribute are defined by the ST_AlgType simple type (§20.1.2.2).

cryptProvider (Cryptographic Provider)	Specifies the cryptographic provider which was used to generate the hash value stored in this document. If the user provided a cryptographic provider which was not the system's built-in provider, then that provider shall be stored here so it can subsequently be used if available.
	
	If this attribute is omitted, then the built-in cryptographic provider on the system shall be used.
	
	[*Example*: Consider a PresentationML document with the following information stored in its protection element:
	
	`<… p:cryptProvider="Krista'sProvider"`
	`p:hashData="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The cryptProvider attribute value of `Krista'sProvider` specifies that the cryptographic provider with name "Krista's Provider" shall be used if available. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

cryptProviderType (Cryptographic Provider Type)	Specifies the kind of cryptographic provider to be used.
	
	[*Example*: Consider a PresentationML document with the following information stored in its protection element:
	
	`<… p:cryptProviderType="rsaAES"`
	`p:hashData="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`

Attributes	Description
	The cryptProviderType attribute value of `rsaAES` specifies that the cryptographic provider type shall be an Advanced Encryption Standard provider. *end example*] The possible values for this attribute are defined by the ST_CryptProv simple type (§20.1.2.4).

cryptProviderTypeExt (Cryptographic Provider Type Extensibility)	Specifies that a cryptographic provider type which was not defined by ISO/IEC 29500 has been used to generate the hash value stored with this document.
	
	This value, when present, shall be interpreted based on the value of the cryptProviderTypeExtSource attribute in order to determine the provider type used, which shall be application-defined. [*Rationale*: This extensibility affords the fact that with exponentially increasing computing power, documents created in the future might need to utilize as yet undefined cryptographic provider types in order to remain secure. *end rationale*]
	
	If this value is present, the cryptProviderType attribute value shall be ignored in favor of the provider type defined by this attribute.
	
	[*Example*: Consider a PresentationML document with the following information stored in its protection element:
	
	`<… p:cryptProviderTypeExt="00A5691D"`
	`p:cryptProvideTypeExtSource="futureCryptography"`
	`p:hashData="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The cryptProviderTypeExt attribute value of `00A5691D` specifies that the provider type associated with hex code `A5691D` shall be used as defined by the `futureCryptography` application. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

cryptProviderTypeExtSource (Provider Type Extensibility Source)	Specifies the application which defined the provider type value specified by the cryptProviderTypeExt attribute.
	
	[*Example*: Consider a PresentationML document with the following information stored in its protection element:
	
	`<… p:cryptProviderTypeExt="00A5691D"`
	`p:cryptProvideTypeExtSource="futureCryptography"`
	`p:hashData="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The cryptProvideTypeExtSource attribute value of `futureCryptography` specifies that the provider type used here was published by the `futureCryptography` application. *end example*]

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

hashData (Password Hash)	Specifies the hash value for the password stored with this document. This value shall be compared with the resulting hash value after hashing the user-supplied password using the algorithm specified by the preceding attributes and parent XML element, and if the two values match, the protection shall no longer be enforced.
	
	If this value is omitted, then no password shall be associated with the protection, and it can be turned off without supplying any password.
	
	[*Example*: Consider a PresentationML document with the following information stored in its protection element:
	
	`<… p:cryptAlgorithmClass="hash"`
	`p:cryptAlgorithmType="typeAny"`
	`p:cryptAlgorithmSid="1"`
	`p:hashData="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The hashData attribute value of `9oN7nWkCAyEZib1RomSJTjmPpCY=` specifies that the user-supplied password shall be hashed using the pre-processing defined by the parent element (if any) followed by the SHA-1 algorithm (specified via the cryptAlgorithmSid attribute value of 1) and that the resulting has value must be `9oN7nWkCAyEZib1RomSJTjmPpCY=` for the protection to be disabled. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.

saltData (Salt for Password Verifier)	Specifies the salt which was prepended to the user-supplied password before it was hashed using the hashing algorithm defined by the preceding attribute values to generate the hashData attribute, and which shall also be prepended to the user-supplied password before attempting to generate a hash value for comparison. A *salt* is a random string which is added to a user-supplied password before it is hashed in order to prevent a malicious party from pre-calculating all possible password/hash combinations and simply using those precalculated values (often referred to as a "dictionary attack").
	
	If this attribute is omitted, then no salt shall be prepended to the user-supplied password before it is hashed for comparison with the stored hash value.
	
	[*Example*: Consider a PresentationML document with the following information stored in its protection element:
	
	`<… p:saltData="ZUdHa+D8F/OAKP3I7ssUnQ=="`
	`p:hashData="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The saltData attribute value of `ZUdHa+D8F/OAKP3I7ssUnQ==` specifies that the user-supplied password shall have this value prepended before it is run through the specified hashing algorithm to generate a resulting hash value for comparison. *end example*]

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.

spinCount (Iterations to Run Hashing Algorithm)	Specifies the number of times the hashing function shall be iteratively run (runs using each iteration's result plus a 4 byte value (0-based, little endian) containing the number of the iteration as the input for the next iteration) when attempting to compare a user-supplied password with the value stored in the hashData attribute. [*Rationale*: Running the algorithm many times increases the cost of exhaustive search attacks correspondingly. Storing this value allows for the number of iterations to be increased over time to accommodate faster hardware (and hence the ability to run more iterations in less time). *end rationale*]
	
	[*Example*: Consider a PresentationML document with the following information stored in its protection element:
	
	`<… p:spinCount="100000"`
	`p:hashData="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The spinCount attribute value of `100000` specifies that the hashing function shall be run one hundred thousand times to generate a hash value for comparison with the hash attribute. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

16.3 Slides

16.3.1 Embedded Objects

16.3.1.1 Additional attribute for control element (Part 1, §19.3.2.1)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description
spid (Embedded object Shape ID)	Specifies the identifier of the shape associated with this Embedded object. The shape contains all coordinate anchoring information. This optional attribute shall be present if the parent element does not contain a child pic element. The possible values for this attribute are defined by the ST_ShapeID simple type (Part 1, §20.1.10.55).

16.3.1.2 Additional attribute for oleObj element (Part 1, §19.3.2.4)

The following additional attributes can be specified for a document of a transitional conformance class:

Attributes	Description
spid (Embedded object Shape ID)	Specifies the identifier of the shape associated with this Embedded object. The shape contains all coordinate anchoring information. This optional attribute shall be present if the parent element does not contain a child pic element. The possible values for this attribute are defined by the ST_ShapeID simple type (Part 1, §20.1.10.55).

16.4 Simple Types

The following additional simple type information in the http://schemas.openxmlformats.org/presentationml/2006/main namespace is used for documents of a transitional conformance class.

16.4.1 ST_WebColorType (HTML Slide Navigation Control Colors)

This simple type specifies the coloring that should be used when outputting to web formats.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

Enumeration Value	Description
blackTextOnWhite (Black Text on White Colors)	Black Text on White coloring should be used.
browser (Browser Colors)	Browser coloring should be used.
none (Non-specific Colors)	No specific coloring has been specified.
presentationAccent (Presentation Accent Colors)	Presentation accent coloring should be used.
presentationText (Presentation Text Colors)	Presentation text coloring should be used.
whiteTextOnBlack (White Text on Black Colors)	White text on black coloring should be used.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_WebColorType) is located in §A.3. *end note*]

16.4.2 ST_WebEncoding (Web Encoding)

This simple type specifies a string representing the HTML character set used when outputting to web formats.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_WebEncoding) is located in §A.3. *end note*]

16.4.3 ST_WebScreenSize (HTML/Web Screen Size Target)

This simple type specifies the intended screen resolution for output to web formats.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

Enumeration Value	Description
1024x768 (HTML/Web Size Enumeration 1024x768)	Screen size is 1024x768 pixels
1152x882 (HTML/Web Size Enumeration 1152x882)	Screen size is 1152x882 pixels
1152x900 (HTML/Web Size Enumeration 1152x900)	Screen size is 1152x900 pixels
1280x1024 (HTML/Web Size Enumeration 1280x1024)	Screen size is 1280x1024 pixels
1600x1200 (HTML/Web Size Enumeration 1600x1200)	Screen size is 1600x1200 pixels
1800x1400 (HTML/Web Size Enumeration 1800x1400)	Screen size is 1800x1400 pixels
1920x1200 (HTML/Web Size Enumeration 1920x1200)	Screen size is 1920x1200 pixels
544x376 (HTML/Web Size Enumeration 544x376)	Screen size is 544x376 pixels
640x480 (HTML/Web Size Enumeration 640x480)	Screen size is 640x480 pixels
720x512 (HTML/Web Size Enumeration 720x515)	Screen size is 720x512 pixels
800x600 (HTML/Web Size Enumeration 800x600)	Screen size is 800x600 pixels

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_WebScreenSize) is located in §A.3. *end note*]

16.5 Changed attributes

The following attributes, which are defined in subclauses within Part 1, §19, "PresentationML", have different source relationships when used in documents of the Transitional conformance class:

16.5.1 Changed attribute for bold element (Part 1, §19.2.1.1)

Attributes	Description
id (Relationship Identifier) Namespace: .../officeDocument/2006/relationships	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location of this embedded font that is referenced in a presentation. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

16.5.2 Changed attribute for boldItalic element (Part 1, §19.2.1.2)

Attributes	Description
id (Relationship Identifier) Namespace: .../officeDocument/2006/relationships	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location of this embedded font that is referenced in a presentation. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

16.5.3 Changed attribute for font element (Part 1, §19.2.1.13)

Attributes	Description
charset (Similar Character Set) Namespace: .../drawingml/2006/main	Specifies the character set that is supported by the parent font. This information can be used in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available. The value of this attribute shall be interpreted as follows:

Value	Description
`0x00`	Specifies the ANSI character set. (IANA name `iso-8859-1`)
`0x01`	Specifies the default character set.
`0x02`	Specifies the Symbol character set. This value specifies that the characters in the Unicode private use area (U+FF00 to U+FFFF) of the font should be used to display characters in the range U+0000 to U+00FF.
`0x4D`	Specifies a Macintosh (Standard Roman) character set. (IANA name `macintosh`)
`0x80`	Specifies the JIS character set. (IANA name `shift_jis`)
`0x81`	Specifies the Hangul character set. (IANA name `ks_c_5601-1987`)
`0x82`	Specifies a Johab character set. (IANA name KS `C-5601-1992`)
`0x86`	Specifies the GB-2312 character set. (IANA name GBK)
`0x88`	Specifies the Chinese Big Five character set. (IANA name `Big5`)
`0xA1`	Specifies a Greek character set. (IANA name `windows-1253`)
`0xA2`	Specifies a Turkish character set. (IANA name `iso-8859-9`)
`0xA3`	Specifies a Vietnamese character set. (IANA name `windows-1258`)
`0xB1`	Specifies a Hebrew character set. (IANA name `windows-1255`)
`0xB2`	Specifies an Arabic character set. (IANA name `windows-1256`)
`0xBA`	Specifies a Baltic character set. (IANA name `windows-1257`)

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>0xCC</td><td>Specifies a Russian character set. (IANA name windows-1251)</td></tr>
<tr><td>0xDE</td><td>Specifies a Thai character set. (IANA name windows-874)</td></tr>
<tr><td>0xEE</td><td>Specifies an Eastern European character set. (IANA name windows-1250)</td></tr>
<tr><td>0xFF</td><td>Specifies an OEM character set not defined by ISO/IEC 29500.</td></tr>
<tr><td>Any other value</td><td>Application-defined, can be ignored.</td></tr>
</table>

The possible values for this attribute are defined by the W3C XML Schema byte datatype.
</td></tr>
<tr><td>panose (Panose Setting)

Namespace: …/drawingml/2006/main</td><td>Specifies the Panose-1 classification number for the current font using the mechanism defined in §5.2.7.17 of ISO/IEC 14496-22.

The possible values for this attribute are defined by the ST_Panose simple type (Part 1, §22.9.2.8).</td></tr>
<tr><td>pitchFamily (Similar Font Family)

Namespace: …/drawingml/2006/main</td><td>Specifies the font pitch as well as the font family for the corresponding font. Because the value of this attribute is determined by a byte variable this value shall be interpreted as follows:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x00</td><td>DEFAULT PITCH + UNKNOWN FONT FAMILY</td></tr>
<tr><td>0x01</td><td>FIXED PITCH + UNKNOWN FONT FAMILY</td></tr>
<tr><td>0x02</td><td>VARIABLE PITCH + UNKNOWN FONT FAMILY</td></tr>
<tr><td>0x10</td><td>DEFAULT PITCH + ROMAN FONT FAMILY</td></tr>
<tr><td>0x11</td><td>FIXED PITCH + ROMAN FONT FAMILY</td></tr>
<tr><td>0x12</td><td>VARIABLE PITCH + ROMAN FONT FAMILY</td></tr>
<tr><td>0x20</td><td>DEFAULT PITCH + SWISS FONT FAMILY</td></tr>
<tr><td>0x21</td><td>FIXED PITCH + SWISS FONT FAMILY</td></tr>
<tr><td>0x22</td><td>VARIABLE PITCH + SWISS FONT FAMILY</td></tr>
<tr><td>0x30</td><td>DEFAULT PITCH + MODERN FONT FAMILY</td></tr>
<tr><td>0x31</td><td>FIXED PITCH + MODERN FONT FAMILY</td></tr>
<tr><td>0x32</td><td>VARIABLE PITCH + MODERN FONT FAMILY</td></tr>
<tr><td>0x40</td><td>DEFAULT PITCH + SCRIPT FONT FAMILY</td></tr>
<tr><td>0x41</td><td>FIXED PITCH + SCRIPT FONT FAMILY</td></tr>
<tr><td>0x42</td><td>VARIABLE PITCH + SCRIPT FONT FAMILY</td></tr>
<tr><td>0x50</td><td>DEFAULT PITCH + DECORATIVE FONT FAMILY</td></tr>
<tr><td>0x51</td><td>FIXED PITCH + DECORATIVE FONT FAMILY</td></tr>
</table>
</td></tr>
</table>

Attributes	Description
	0x52 | VARIABLE PITCH + DECORATIVE FONT FAMILY This information is determined by querying the font when present and shall not be modified when the font is not available. This information can be used in font substitution logic to locate an appropriate substitute font when this font is not available. The possible values for this attribute are defined by the W3C XML Schema byte datatype.
typeface (Text Typeface) Namespace: .../drawingml/2006/main	Specifies the typeface, or name of the font that is to be used. The typeface is a string name of the specific font that should be used in rendering the presentation. If this font is not available within the font list of the generating application than font substitution logic should be utilized in order to select an alternate font. The possible values for this attribute are defined by the ST_TextTypeface simple type (Part 1, §20.1.10.81).

16.5.4 Changed attribute for handoutMasterId element (Part 1, §19.2.1.14)

Attributes	Description
id (Relationship Identifier) Namespace: .../officeDocument/2006/relationships	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location within a presentation of the handoutMaster element defining this handout master. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

16.5.5 Changed attribute for italic element (Part 1, §19.2.1.16)

Attributes	Description
id (Relationship Identifier) Namespace: .../officeDocument/2006/relationships	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location of this embedded font that is referenced in a presentation. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

16.5.6 Changed attribute for notesMasterId element (Part 1, §19.2.1.20)

Attributes	Description
id (Relationship Identifier) Namespace: .../officeDocument/2006/relationships	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location within a presentation of the notesMaster element defining this notes master. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

16.5.7 Changed attribute for notesSz element (Part 1, §19.2.1.22)

Attributes	Description
cx (Extent Length) Namespace: .../drawingml/2006/main	Specifies the length of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `<… cx="1828800" cy="200000"/>` The cx attributes specifies that this object has a height of `1828800` EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (Part 1, §20.1.10.42).
cy (Extent Width) Namespace: .../drawingml/2006/main	Specifies the width of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `< … cx="1828800" cy="200000"/>` The cy attribute specifies that this object has a width of `200000` EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (Part 1, §20.1.10.42).

16.5.8 Changed attribute for regular element (Part 1, §19.2.1.29)

Attributes	Description
id (Relationship Identifier) Namespace: .../officeDocument/2006/relationships	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location of this embedded font that is referenced in a presentation. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

16.5.9 Changed attribute for sld element (Part 1, §19.2.1.31)

Attributes	Description
id (Relationship ID) Namespace: .../officeDocument/2006/relationships	This attribute specifies the relationship id that is used to reference to the actual slide XML file that contains all the information to the slide listed within the slide list. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

16.5.10 Changed attribute for sldId element (Part 1, §19.2.1.33)

Attributes	Description
id (Relationship Identifier) Namespace: .../officeDocument/2006/relationships	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location within a presentation of the sld element defining this slide. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

16.5.11 Changed attribute for sldMasterId element (Part 1, §19.2.1.36)

Attributes	Description
id (Relationship Identifier) Namespace: .../officeDocument/2006/relationships	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location within a presentation of the sldMaster element defining this slide master. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

16.5.12 Changed attribute for SmartTags element (Part 1, §19.2.1.40)

Attributes	Description
id (Relationship Identifier) Namespace: .../officeDocument/2006/relationships	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location of this smart tag. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

16.5.13 Changed attribute for gridSpacing element (Part 1, §19.2.2.3)

Attributes	Description
cx (Extent Length) Namespace: .../drawingml/2006/main	Specifies the length of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `<… cx="1828800" cy="200000"/>` The cx attributes specifies that this object has a height of `1828800` EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (Part 1, §20.1.10.42).
cy (Extent Width) Namespace: .../drawingml/2006/main	Specifies the width of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `< … cx="1828800" cy="200000"/>` The cy attribute specifies that this object has a width of `200000` EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (Part 1, §20.1.10.42).

16.5.14 Changed attribute for origin element (Part 1, §19.2.2.9)

Attributes	Description
x (X-Axis Coordinate) Namespace: …/drawingml/2006/main	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The x attribute defines an x-coordinate of `0`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (Part 1, §20.1.10.16).
y (Y-Axis Coordinate) Namespace: …/drawingml/2006/main	Specifies a coordinate on the y-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The y attribute defines a y-coordinate of `100`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (Part 1, §20.1.10.16).

16.5.15 Changed attribute for sld element (Part 1, §19.2.2.14)

Attributes	Description
id (Relationship Identifier) Namespace: …/officeDocument/2006/relationships	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location of this presentation slide within a presentation. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

16.5.16 Changed attribute for bgRef element (Part 1, §19.3.1.3)

Attributes	Description
idx (Style Matrix Index) Namespace: .../drawingml/2006/main	Specifies the style matrix index of the style referred to. The possible values for this attribute are defined by the ST_StyleMatrixColumnIndex simple type (Part 1, §20.1.10.57).

16.5.17 Changed attribute for blipFill element (Part 1, §19.3.1.4)

Attributes	Description
dpi (DPI Setting) Namespace: .../drawingml/2006/main	Specifies the DPI (dots per inch) used to calculate the size of the blip. If not present or zero, the DPI in the blip is used. [*Note*: This attribute is primarily used to keep track of the picture quality within a document. There are different levels of quality needed for print than on-screen viewing and thus a need to track this information. *end note*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
rotWithShape (Rotate With Shape) Namespace: .../drawingml/2006/main	Specifies that the fill should rotate with the shape. That is, when the shape that has been filled with a picture and the containing shape (say a rectangle) is transformed with a rotation then the fill is transformed with the same rotation. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

16.5.18 Changed attribute for clrMap element (Part 1, §19.3.1.6)

Attributes	Description
accent1 (Accent 1) Namespace: .../drawingml/2006/main	Specifies a color defined which is associated as the accent 1 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).
accent2 (Accent 2) Namespace: .../drawingml/2006/main	Specifies a color defined which is associated as the accent 2 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).
accent3 (Accent 3) Namespace: .../drawingml/2006/main	Specifies a color defined which is associated as the accent 3 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).

Attributes	Description
accent4 (Accent 4) Namespace: …/drawingml/2006/main	Specifies a color defined which is associated as the accent 4 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).
accent5 (Accent 5) Namespace: …/drawingml/2006/main	Specifies a color defined which is associated as the accent 5 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).
accent6 (Accent 6) Namespace: …/drawingml/2006/main	Specifies a color defined which is associated as the accent 6 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).
bg1 (Background 1) Namespace: …/drawingml/2006/main	A color defined which is associated as the first background color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).
bg2 (Background 2) Namespace: …/drawingml/2006/main	Specifies a color defined which is associated as the second background color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).
folHlink (Followed Hyperlink) Namespace: …/drawingml/2006/main	Specifies a color defined which is associated as the color for a followed hyperlink. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).
hlink (Hyperlink) Namespace: …/drawingml/2006/main	Specifies a color defined which is associated as the color for a hyperlink. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).
tx1 (Text 1) Namespace: …/drawingml/2006/main	Specifies a color defined which is associated as the first text color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).

Attributes	Description
tx2 (Text 2) Namespace: …/drawingml/2006/main	Specifies a color defined which is associated as the second text color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).

16.5.19 Changed attribute for cNvPicPr element (Part 1, §19.3.1.11)

Attributes	Description
preferRelativeResize (Relative Resize Preferred) Namespace: …/drawingml/2006/main	Specifies if the user interface should show the resizing of the picture based on the picture's current size or its original size. If this attribute is set to true, then scaling is relative to the original picture size as opposed to the current picture size. [*Example*: Consider the case where a picture has been resized within a document and is now 50% of the originally inserted picture size. Now if the user chooses to make a later adjustment to the size of this picture within the generating application, then the value of this attribute should be checked. If this attribute is set to true then a value of 50% is shown. Similarly, if this attribute is set to false, then a value of 100% should be shown because the picture has not yet been resized from its current (smaller) size. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

16.5.20 Changed attribute for cNvPr element (Part 1, §19.3.1.12)

Attributes	Description
descr (Alternative Text for Object) Namespace: …/drawingml/2006/main	Specifies alternative text for the current DrawingML object, for use by assistive technologies or applications that do not display the current object. If this element is omitted, then no alternative text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: `<… descr="A picture of a bowl of fruit">` The descr attribute contains alternative text that can be used in place of the actual DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

Attributes	Description
hidden (Hidden) Namespace: …/drawingml/2006/main	Specifies whether this DrawingML object is displayed. When a DrawingML object is displayed within a document, that object can be hidden (i.e., present, but not visible). This attribute determines whether the object is rendered or made hidden. [*Note*: An application can have settings which allow this object to be viewed. *end note*] If this attribute is omitted, then the parent DrawingML object shall be displayed (i.e., not hidden). [*Example*: Consider an inline DrawingML object that must be hidden within the document's content. This setting would be specified as follows: `<… hidden="true" />` The hidden attribute has a value of `true`, which specifies that the DrawingML object is hidden and not displayed when the document is displayed. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Unique Identifier) Namespace: …/drawingml/2006/main	Specifies a unique identifier for the current DrawingML object within the current document. This ID can be used to assist in uniquely identifying this object so that it can be referred to by other parts of the document. If multiple objects within the same document share the same id attribute value, then the document shall be considered non-conformant. [*Example*: Consider a DrawingML object defined as follows: `<… id="10" … >` The id attribute has a value of `10`, which is the unique identifier for this DrawingML object. *end example*] The possible values for this attribute are defined by the ST_DrawingElementId simple type (Part 1, §20.1.10.21).
name (Name) Namespace: …/drawingml/2006/main	Specifies the name of the object. [*Note*: Typically, this is used to store the original file name of a picture object. *end note*] [*Example*: Consider a DrawingML object defined as follows: `< … name="foo.jpg" >` The name attribute has a value of `foo.jpg`, which is the name of this DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string

Attributes	Description
	datatype.
title (Title) Namespace: …/drawingml/2006/main	Specifies the title (caption) of the current DrawingML object. If this attribute is omitted, then no title text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: `<… title="Process Flow Diagram">` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

16.5.21 Changed attribute for cNvSpPr element (Part 1, §19.3.1.13)

Attributes	Description
txBox (Text Box) Namespace: …/drawingml/2006/main	Specifies that the corresponding shape is a text box and thus should be treated as such by the generating application. If this attribute is omitted then it is assumed that the corresponding shape is not specifically a text box. [*Note*: Because a shape is not specified to be a text box does not mean that it cannot have text attached to it. A text box is merely a specialized shape with specific properties. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

16.5.22 Changed attribute for contentPart element (Part 1, §19.3.1.14)

Attributes	Description
id (Relationship to Part) Namespace: …/officeDocument/2006/relationships	Specifies the relationship ID to a content part. [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

16.5.23 Changed attribute for custData element (Part 1, §19.3.1.17)

Attributes	Description
id (Relationship ID) Namespace: .../officeDocument/2006/relationships	This attribute specifies the relationship id for referencing other resources outside the scope of the current PresentationML file. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

16.5.24 Changed attribute for grpSpPr element (Part 1, §19.3.1.23)

Attributes	Description
bwMode (Black and White Mode) Namespace: .../drawingml/2006/main	Specifies that the group shape should be rendered using only black and white coloring. That is the coloring information for the group shape should be converted to either black or white when rendering the corresponding shapes. No gray is to be used in rendering this image, only stark black and stark white. [*Note*: This does not mean that the group shapes themselves are stored with only black and white color information. This attribute instead sets the rendering mode that the shapes use when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (Part 1, §20.1.10.10).

16.5.25 Changed attribute for sldLayoutId element (Part 1, §19.3.1.40)

Attributes	Description
id (ID Tag) Namespace: .../officeDocument/2006/relationships	Specifies the relationship id value that the generating application can use to resolve which slide layout is used in the creation of the slide. This relationship id is used within the relationship file for the master slide to expose the location of the corresponding layout file within the presentation. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

16.5.26 Changed attribute for spPr element (Part 1, §19.3.1.44)

Attributes	Description
bwMode (Black and White Mode) Namespace: .../drawingml/2006/main	Specifies that the picture should be rendered using only black and white coloring. That is the coloring information for the picture should be converted to either black or white when rendering the picture. No gray is to be used in rendering this image, only stark black and stark white. [*Note*: This does not mean that the picture itself that is stored within the file is necessarily a black and white picture. This attribute instead sets the rendering mode that

Attributes	Description
	the picture has applied to when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (Part 1, §20.1.10.10).

16.5.27 Changed attribute for tags element (Part 1, §19.3.1.47)

Attributes	Description
id (Relationship ID) Namespace: .../officeDocument/2006/relationships	This attribute specifies the relationship identifier for the customer data tag. This allows for a link to a resource that is external from the current XML document but still contained within the presentation document. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

16.5.28 Changed attribute for xfrm element (Part 1, §19.3.1.53)

Attributes	Description
flipH (Horizontal Flip) Namespace: .../drawingml/2006/main	Specifies a horizontal flip. When true, this attribute defines that the shape is flipped horizontally about the center of its bounding box. [*Example*: The following illustrates the effect of a horizontal flip.  *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
flipV (Vertical Flip) Namespace: .../drawingml/2006/main	Specifies a vertical flip. When true, this attribute defines that the group is flipped vertically about the center of its bounding box. [*Example*: The following illustrates the effect of a vertical flip.  *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

Attributes	Description
rot (Rotation) Namespace: .../drawingml/2006/main	Specifies the rotation of the Graphic Frame. The units for which this attribute is specified in reside within the simple type definition referenced below. The possible values for this attribute are defined by the ST_Angle simple type (Part 1, §20.1.10.3).

16.5.29 Changed attribute for control element (Part 1, §19.3.2.1)

Attributes	Description
id (Relationship ID) Namespace: .../officeDocument/2006/relationships	Specifies the relationship id that is used to identify this Embedded object from within a slide. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

16.5.30 Changed attribute for oleObj element (Part 1, §19.3.2.4)

Attributes	Description
id (Relationship ID) Namespace: .../officeDocument/2006/relationships	Specifies the relationship id that is used to identify this Embedded object from within a slide. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

16.5.31 Changed attribute for pos element (Part 1, §19.4.5)

Attributes	Description
x (X-Axis Coordinate) Namespace: .../drawingml/2006/main	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The x attribute defines an x-coordinate of `0`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (Part 1, §20.1.10.16).
y (Y-Axis Coordinate) Namespace: .../drawingml/200	Specifies a coordinate on the y-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object:

Attributes	Description
6/main	`<… x="0" y="100" />` The y attribute defines a y-coordinate of `100`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (Part 1, §20.1.10.16).

16.5.32 Changed attribute for snd element (Part 1, §19.5.68)

Attributes	Description
embed (Embedded Audio File Relationship ID) Namespace: …/officeDocument/2006/relationships	Specifies the identification information for an embedded audio file. This attribute is used to specify the location of an object that resides locally within the file. [*Note*: A list of suggested audio types is provided in Part 1, §15.2.2. *end note*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).
name (Sound Name) Namespace: …/drawingml/2006/main	Specifies the original name or given short name for the corresponding sound. This is used to distinguish this sound from others by providing a human readable name for the attached sound should the user need to identify the sound among others within the UI. The possible values for this attribute are defined by the W3C XML Schema string datatype.

16.5.33 Changed attribute for sndTgt element (Part 1, §19.5.70)

Attributes	Description
embed (Embedded Audio File Relationship ID) Namespace: …/officeDocument/2006/relationships	Specifies the identification information for an embedded audio file. This attribute is used to specify the location of an object that resides locally within the file. [*Note*: A list of suggested audio types is provided in Part 1, §15.2.2. *end note*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).
name (Sound Name) Namespace: …/drawingml/2006/main	Specifies the original name or given short name for the corresponding sound. This is used to distinguish this sound from others by providing a human readable name for the attached sound should the user need to identify the sound among others within the UI. The possible values for this attribute are defined by the W3C XML Schema string datatype.

17. DrawingML - Framework Reference Material

[*Note*: For further information on the mapping of elements and attributes to OPC parts, see the Bibliography entry, "Information on elements, attributes, and OPC parts in ISO/IEC 29500 (OOXML)". *end note*]

17.1 DrawingML - Main

17.1.1 Table of Contents

This subclause is informative.

End of informative text.

17.1.2 Simple Types

The following additional simple type information in the http://schemas.openxmlformats.org/drawingml/2006/main namespace is used for documents of a transitional conformance class.

17.1.2.1 Additional member types for the union in ST_FixedPercentage (Part 1, §20.1.10.24)

The value space of the following additional member types can be used within the context of this simple type for a document of a transitional conformance class.

- The ST_FixedPercentageDecimal simple type (§17.1.2.7).

17.1.2.2 Additional member types for the union in ST_Percentage (Part 1, §20.1.10.40)

The value space of the following additional member types can be used within the context of this simple type for a document of a transitional conformance class.

- The ST_PercentageDecimal simple type (Part 4, §12.1.2.12).

17.1.2.3 Additional member types for the union in ST_PositiveFixedPercentage (Part 1, §20.1.10.45)

The value space of the following additional member types can be used within the context of this simple type for a document of a transitional conformance class.

- The ST_PositiveFixedPercentageDecimal simple type (§17.1.2.8).

17.1.2.4 Additional member types for the union in ST_PositivePercentage (Part 1, §20.1.10.46)

The value space of the following additional member types can be used within the context of this simple type for a document of a transitional conformance class.

- The ST_PositivePercentageDecimal simple type (§17.1.2.9).

17.1.2.5 Additional member types for the union in ST_TextFontScalePercentOrPercentString (Part 1, §20.1.10.67)

The value space of the following additional member types can be used within the context of this simple type for a document of a transitional conformance class.

- The ST_TextFontScalePercent simple type (§17.1.2.10).

17.1.2.6 Additional member types for the union in ST_TextSpacingPercentOrPercentString (Part 1, §20.1.10.77)

The value space of the following additional member types can be used within the context of this simple type for a document of a transitional conformance class.

- The ST_TextSpacingPercent simple type (§17.1.2.11).

17.1.2.7 ST_FixedPercentageDecimal (Fixed Percentage)

This simple type represents a fixed percentage in 1000ths of a percent. Range from [-100%, 100%].

This simple type's contents are a restriction of the ST_PercentageDecimal datatype (Part 1, §20.1.10.41).

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to -100000.

- This simple type has a maximum value of less than or equal to 100000.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FixedPercentageDecimal) is located in §A.4.1. *end note*]

17.1.2.8 ST_PositiveFixedPercentageDecimal (Positive Fixed Percentage)

This simple type represents a positive fixed percentage in 1000ths of a percent. Range from [0%, 100%].

This simple type's contents are a restriction of the ST_PercentageDecimal datatype (Part 1, §20.1.10.41).

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 0.
- This simple type has a maximum value of less than or equal to 100000.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PositiveFixedPercentageDecimal) is located in §A.4.1. *end note*]

17.1.2.9 ST_PositivePercentageDecimal (Positive Percentage as Decimal Number)

This simple type represents a positive percentage in 1000ths of a percent. Range from 0% up to and including infinity.

This simple type's contents are a restriction of the ST_PercentageDecimal datatype (Part 1, §20.1.10.41).

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 0.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PositivePercentageDecimal) is located in §A.4.1. *end note*]

17.1.2.10 ST_TextFontScalePercent (Text Font Scale Percentage)

This simple type specifies the percentage range text can be scaled to in order to fit, in 1000ths of a percent.

This simple type's contents are a restriction of the ST_PercentageDecimal datatype (Part 1, §20.1.10.41).

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 1000.
- This simple type has a maximum value of less than or equal to 100000.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextFontScalePercent) is located in §A.4.1. *end note*]

17.1.2.11 ST_TextSpacingPercent (Text Spacing Percent)

This type specifies the range of text spacing in thousandths of a percent, in terms of a line.

This simple type's contents are a restriction of the ST_PercentageDecimal datatype (Part 1, §20.1.10.41).

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 0.
- This simple type has a maximum value of less than or equal to 13200000.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextSpacingPercent) is located in §A.4.1. *end note*]

17.1.2.12 ST_PercentageDecimal (Percentage as Decimal Number)

This simple type represents a percentage in 1000ths of a percent, e.g., a value of 1 represents 0.001% == 0.00001; a value of 100000 is equal to 100%. Percentages have no intrinsic units, but are used to scale other values with units.

This simple type's contents are a restriction of the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PercentageDecimal) is located in §A.4.1. *end note*]

17.1.2.13 Additional member types for the union in ST_PrSetCustVal (Part 1, §21.4.7.66)

The value space of the following additional member types can be used within the context of this simple type for a document of a transitional conformance class.

The W3C XML Schema int datatype.

17.1.2.14 ST_TextBulletSizeDecimal (Bullet Size Percentage)

This simple type specifies the range that the bullet percent can be. A bullet percent is the size of the bullet with respect to the text that should follow it. 25000 = 25%, 400000 = 400%

This simple type's contents are a restriction of the ST_PercentageDecimal datatype (Part 4, §12.1.2.12).

This simple type also specifies the following restrictions:

This simple type has a minimum value of greater than or equal to 25000.
This simple type has a maximum value of less than or equal to 400000.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextBulletSizeDecimal) is located in §A.4.1. *end note*]

17.1.2.15 Additional member types for the union in ST_TextBulletSize (Part 1, §20.1.10.86)

The value space of the following additional member types can be used within the context of this simple type for a document of a transitional conformance class.

The ST_TextBulletSizeDecimal simple type (Part 4, §12.1.2.14).

17.2 DrawingML - Legacy Compatibility

Within the context of DrawingML, it shall be possible (for considerations to legacy compatibility) to be able to include explicit references to specific shapes within VML Drawing parts.

[*Example*: A VML Drawing part is used to define ink on a PresentationML slide, but the resulting ink is referenced from the slide by its shape ID using the elements of this namespace. *end example*]

17.2.1 Table of Contents

This subclause is informative.

End of informative text.

17.2.2 Basics

Legacy Compatibility is part of the shape definitions and properties of the DrawingML framework.

17.2.2.1 legacyDrawing (Legacy Drawing Object)

This element specifies the shape ID for a legacy drawing object. These legacy drawing objects all have a shape ID associated with them that is unique across the entire document. In order to store these legacy shape IDs as well as new shape IDs this legacyDrawing element should be used.

Attributes	Description
spid (Shape ID)	Legacy Shape ID that is unique throughout the entire document. Legacy shape IDs should be assigned based on which portion of the document the drawing resides on. The assignment of these ids is broken down into clusters of 1024 values. The first cluster is 1-1024, the second 1025-2048 and so on. This optional attribute shall be present if the parent element does not contain a child picture element. [*Example*: Within a word processing application the spid should be assigned based on the page that the drawing resides on. If the drawing resides on the second page then the assigned spid should be a value between 1025 and 2048. *end example*] [*Example*: Within a spreadsheet application the spid should be assigned based on the sheet that the drawing resides on. If the drawing resides on the second sheet then the assigned spid should be a value between 1025 and 2048. *end example*] [*Example*: Within a presentation application the spid should be assigned based on the slide that the drawing resides on. If the drawing resides on the second slide then the assigned spid should be a value between 1025 and 2048. *end example*] The possible values for this attribute are defined by the ST_ShapeID simple type (Part 1, §20.1.10.55).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Compat) is located in §!ϕMϕ *end note*]

17.3 Changed attributes

The following attributes, which are defined in subclauses within Part 1, §20, "DrawingML - Framework Reference Material", have different source relationships when used in documents of the Transitional conformance class:

17.3.1 Changed attribute for hlinkHover element (Part 1, §20.1.2.2.23)

Attributes	Description
id (Drawing Object Hyperlink Target) Namespace: .../officeDocument/2006/relationships	Specifies the relationship id that when looked up in this slides relationship file contains the target of this hyperlink. This attribute cannot be omitted. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

17.3.2 Changed attribute for snd element (Part 1, §20.1.2.2.32)

Attributes	Description
embed (Embedded Audio File Relationship ID) Namespace: .../officeDocument/2006/relationships	Specifies the identification information for an embedded audio file. This attribute is used to specify the location of an object that resides locally within the file. [*Note*: A list of suggested audio types is provided in Part 1, §15.2.2. *end note*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

17.3.3 Changed attribute for audioFile element (Part 1, §20.1.3.2)

Attributes	Description
link (Linked Relationship ID) Namespace: .../officeDocument/2006/relationships	Specifies the identification information for a linked object. This attribute is used to specify the location of an object that does not reside within this file. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

17.3.4 Changed attribute for quickTimeFile element (Part 1, §20.1.3.4)

Attributes	Description
link (Linked Relationship ID) Namespace: .../officeDocument/2006/relationships	Specifies the identification information for a linked object. This attribute is used to specify the location of an object that does not reside within this file. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

17.3.5 Changed attribute for videoFile element (Part 1, §20.1.3.6)

Attributes	Description
link (Linked Relationship ID) Namespace: .../officeDocument/2006/relationships	Specifies the identification information for a linked object. This attribute is used to specify the location of an object that does not reside within this file. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

17.3.6 Changed attribute for wavAudioFile element (Part 1, §20.1.3.7)

Attributes	Description
embed (Embedded Audio File Relationship ID) Namespace: .../officeDocument/2006/relationships	Specifies the identification information for an embedded audio file. This attribute is used to specify the location of an object that resides locally within the file. [*Note*: A list of suggested audio types is provided in Part 1, §15.2.2. *end note*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

17.3.7 Changed attribute for blip element (Part 1, §20.1.8.13)

Attributes	Description
embed (Embedded Picture Reference) Namespace: .../officeDocument/2006/relationships	Specifies the identification information for an embedded picture. This attribute is used to specify an image that resides locally within the file. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

Attributes	Description
link (Linked Picture Reference) Namespace: .../officeDocument/2006/relationships	Specifies the identification information for a linked picture. This attribute is used to specify an image that does not reside within this file. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

17.3.8 Changed attribute for blipFill element (Part 1, §20.2.2.1)

Attributes	Description
dpi (DPI Setting) Namespace: .../drawingml/2006/main	Specifies the DPI (dots per inch) used to calculate the size of the blip. If not present or zero, the DPI in the blip is used. [*Note*: This attribute is primarily used to keep track of the picture quality within a document. There are different levels of quality needed for print than on-screen viewing and thus a need to track this information. *end note*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
rotWithShape (Rotate With Shape) Namespace: .../drawingml/2006/main	Specifies that the fill should rotate with the shape. That is, when the shape that has been filled with a picture and the containing shape (say a rectangle) is transformed with a rotation then the fill is transformed with the same rotation. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

17.3.9 Changed attribute for cNvPicPr element (Part 1, §20.2.2.2)

Attributes	Description
preferRelativeResize (Relative Resize Preferred) Namespace: .../drawingml/2006/main	Specifies if the user interface should show the resizing of the picture based on the picture's current size or its original size. If this attribute is set to true, then scaling is relative to the original picture size as opposed to the current picture size. [*Example*: Consider the case where a picture has been resized within a document and is now 50% of the originally inserted picture size. Now if the user chooses to make a later adjustment to the size of this picture within the generating application, then the value of this attribute should be checked. If this attribute is set to true then a value of 50% is shown. Similarly, if this attribute is set to false, then a value of 100% should be shown because the picture has not yet been resized from its current (smaller) size. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

17.3.10 Changed attribute for cNvPr element (Part 1, §20.2.2.3)

Attributes	Description
descr (Alternative Text for Object) Namespace: …/drawingml/2006/main	Specifies alternative text for the current DrawingML object, for use by assistive technologies or applications that do not display the current object. If this element is omitted, then no alternative text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: `<… descr="A picture of a bowl of fruit">` The descr attribute contains alternative text that can be used in place of the actual DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
hidden (Hidden) Namespace: …/drawingml/2006/main	Specifies whether this DrawingML object is displayed. When a DrawingML object is displayed within a document, that object can be hidden (i.e., present, but not visible). This attribute determines whether the object is rendered or made hidden. [*Note*: An application can have settings that allow this object to be viewed. *end note*] If this attribute is omitted, then the parent DrawingML object shall be displayed (i.e., not hidden). [*Example*: Consider an inline DrawingML object that must be hidden within the document's content. This setting would be specified as follows: `<… hidden="true" />` The hidden attribute has a value of `true`, which specifies that the DrawingML object is hidden and not displayed when the document is displayed. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Unique Identifier) Namespace: …/drawingml/2006/main	Specifies a unique identifier for the current DrawingML object within the current document. This ID can be used to assist in uniquely identifying this object so that it can be referred to by other parts of the document. If multiple objects within the same document share the same id attribute value, then the document shall be considered non-conformant. [*Example*: Consider a DrawingML object defined as follows: `<… id="10" … >` The id attribute has a value of `10`, which is the unique identifier for this DrawingML

Attributes	Description
	object. *end example*] The possible values for this attribute are defined by the ST_DrawingElementId simple type (Part 1, §20.1.10.21).
name (Name) Namespace: …/drawingml/2006/main	Specifies the name of the object. [*Note*: Typically, this is used to store the original file name of a picture object. *end note*] [*Example*: Consider a DrawingML object defined as follows: `< … name="foo.jpg" >` The name attribute has a value of `foo.jpg`, which is the name of this DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
title (Title) Namespace: …/drawingml/2006/main	Specifies the title (caption) of the current DrawingML object. If this attribute is omitted, then no title text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: `<… title="Process Flow Diagram">` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

17.3.11 Changed attribute for spPr element (Part 1, §20.2.2.6)

Attributes	Description
bwMode (Black and White Mode) Namespace: …/drawingml/2006/main	Specifies that the picture should be rendered using only black and white coloring. That is the coloring information for the picture should be converted to either black or white when rendering the picture. No gray is to be used in rendering this image, only stark black and stark white. [*Note*: This does not mean that the picture itself that is stored within the file is necessarily a black and white picture. This attribute instead sets the rendering mode that the picture has applied to when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (Part 1, §20.1.10.10).

17.3.12 Changed attribute for docPr element (Part 1, §20.4.2.5)

Attributes	Description
descr (Alternative Text for Object) Namespace: …/drawingml/2006/main	Specifies alternative text for the current DrawingML object, for use by assistive technologies or applications that do not display the current object. If this element is omitted, then no alternative text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: `<… descr="A picture of a bowl of fruit">` The descr attribute contains alternative text that can be used in place of the actual DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
hidden (Hidden) Namespace: …/drawingml/2006/main	Specifies whether this DrawingML object is displayed. When a DrawingML object is displayed within a document, that object can be hidden (i.e., present, but not visible). This attribute determines whether the object is rendered or made hidden. [*Note*: An application can have settings that allow this object to be viewed. *end note*] If this attribute is omitted, then the parent DrawingML object shall be displayed (i.e., not hidden). [*Example*: Consider an inline DrawingML object that must be hidden within the document's content. This setting would be specified as follows: `<… hidden="true" />` The hidden attribute has a value of `true`, which specifies that the DrawingML object is hidden and not displayed when the document is displayed. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Unique Identifier) Namespace: …/drawingml/2006/main	Specifies a unique identifier for the current DrawingML object within the current document. This ID can be used to assist in uniquely identifying this object so that it can be referred to by other parts of the document. If multiple objects within the same document share the same id attribute value, then the document shall be considered non-conformant. [*Example*: Consider a DrawingML object defined as follows: `<… id="10" … >` The id attribute has a value of `10`, which is the unique identifier for this DrawingML

Attributes	Description
	object. *end example*] The possible values for this attribute are defined by the ST_DrawingElementId simple type (Part 1, §20.1.10.21).
name (Name) Namespace: …/drawingml/2006/main	Specifies the name of the object. [*Note*: Typically, this is used to store the original file name of a picture object. *end note*] [*Example*: Consider a DrawingML object defined as follows: `< … name="foo.jpg" >` The name attribute has a value of `foo.jpg`, which is the name of this DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
title (Title) Namespace: …/drawingml/2006/main	Specifies the title (caption) of the current DrawingML object. If this attribute is omitted, then no title text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: `<… title="Process Flow Diagram">` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

17.3.13 Changed attribute for extent element (Part 1, §20.4.2.7)

Attributes	Description
cx (Extent Width) Namespace: …/drawingml/2006/main	Specifies the width of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `<… cx="1828800" cy="200000"/>` The cx attribute specifies that this object has a width of `1828800` EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (Part 1, §20.1.10.42).

Attributes	Description
cy (Extent Height) Namespace: …/drawingml/2006/main	Specifies the height of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `< … cx="1828800" cy="200000"/>` The cy attribute specifies that this object has a height of `200000` EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (Part 1, §20.1.10.42).

17.3.14 Changed attribute for lineTo element (Part 1, §20.4.2.9)

Attributes	Description
x (X-Axis Coordinate) Namespace: …/drawingml/2006/main	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The x attribute defines an x-coordinate of `0`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (Part 1, §20.1.10.16).
y (Y-Axis Coordinate) Namespace: …/drawingml/2006/main	Specifies a coordinate on the y-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The y attribute defines a y-coordinate of `100`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (Part 1, §20.1.10.16).

17.3.15 Changed attribute for simplePos element (Part 1, §20.4.2.13)

Attributes	Description
x (X-Axis Coordinate)	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified by the parent XML element.

Attributes	Description
Namespace: …/drawingml/2006/main	[*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The x attribute defines an x-coordinate of `0`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (Part 1, §20.1.10.16).
y (Y-Axis Coordinate) Namespace: …/drawingml/2006/main	Specifies a coordinate on the y-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The y attribute defines a y-coordinate of `100`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (Part 1, §20.1.10.16).

17.3.16 Changed attribute for start element (Part 1, §20.4.2.14)

Attributes	Description
x (X-Axis Coordinate) Namespace: …/drawingml/2006/main	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The x attribute defines an x-coordinate of `0`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (Part 1, §20.1.10.16).

Attributes	Description
y (Y-Axis Coordinate) Namespace: …/drawingml/2006/main	Specifies a coordinate on the y-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The y attribute defines a y-coordinate of 100. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (Part 1, §20.1.10.16).

17.3.17 Changed attribute for blipFill element (Part 1, §20.5.2.2)

Attributes	Description
dpi (DPI Setting) Namespace: …/drawingml/2006/main	Specifies the DPI (dots per inch) used to calculate the size of the blip. If not present or zero, the DPI in the blip is used. [*Note*: This attribute is primarily used to keep track of the picture quality within a document. There are different levels of quality needed for print than on-screen viewing and thus a need to track this information. *end note*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
rotWithShape (Rotate With Shape) Namespace: …/drawingml/2006/main	Specifies that the fill should rotate with the shape. That is, when the shape that has been filled with a picture and the containing shape (say a rectangle) is transformed with a rotation then the fill is transformed with the same rotation. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

17.3.18 Changed attribute for cNvPicPr element (Part 1, §20.5.2.7)

Attributes	Description
preferRelativeResize (Relative Resize Preferred) Namespace: …/drawingml/2006/main	Specifies if the user interface should show the resizing of the picture based on the picture's current size or its original size. If this attribute is set to true, then scaling is relative to the original picture size as opposed to the current picture size. [*Example*: Consider the case where a picture has been resized within a document and is now 50% of the originally inserted picture size. Now if the user chooses to make a later adjustment to the size of this picture within the generating application, then the value of this attribute should be checked. If this attribute is set to true then a value of 50% is shown. Similarly, if this attribute is set to false, then a value of 100% should be shown because the picture has not yet been

Attributes	Description
	resized from its current (smaller) size. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

17.3.19 Changed attribute for cNvPr element (Part 1, §20.5.2.8)

Attributes	Description
descr (Alternative Text for Object) Namespace: …/drawingml/2006/main	Specifies alternative text for the current DrawingML object, for use by assistive technologies or applications that do not display the current object. If this element is omitted, then no alternative text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: `<… descr="A picture of a bowl of fruit">` The descr attribute contains alternative text that can be used in place of the actual DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
hidden (Hidden) Namespace: …/drawingml/2006/main	Specifies whether this DrawingML object is displayed. When a DrawingML object is displayed within a document, that object can be hidden (i.e., present, but not visible). This attribute determines whether the object is rendered or made hidden. [*Note*: An application can have settings that allow this object to be viewed. *end note*] If this attribute is omitted, then the parent DrawingML object shall be displayed (i.e., not hidden). [*Example*: Consider an inline DrawingML object that must be hidden within the document's content. This setting would be specified as follows: `<… hidden="true" />` The hidden attribute has a value of `true`, which specifies that the DrawingML object is hidden and not displayed when the document is displayed. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

Attributes	Description
id (Unique Identifier) Namespace: …/drawingml/2006/main	Specifies a unique identifier for the current DrawingML object within the current document. This ID can be used to assist in uniquely identifying this object so that it can be referred to by other parts of the document. If multiple objects within the same document share the same id attribute value, then the document shall be considered non-conformant. [*Example*: Consider a DrawingML object defined as follows: `<… id="10" … >` The id attribute has a value of `10`, which is the unique identifier for this DrawingML object. *end example*] The possible values for this attribute are defined by the ST_DrawingElementId simple type (Part 1, §20.1.10.21).
name (Name) Namespace: …/drawingml/2006/main	Specifies the name of the object. [*Note*: Typically, this is used to store the original file name of a picture object. *end note*] [*Example*: Consider a DrawingML object defined as follows: `< … name="foo.jpg" >` The name attribute has a value of `foo.jpg`, which is the name of this DrawingML object. *end example*] The possible values for this attribute are defined by the type W3C XML Schema string.
title (Title) Namespace: …/drawingml/2006/main	Specifies the title (caption) of the current DrawingML object. If this attribute is omitted, then no title text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: `<… title="Process Flow Diagram">` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

17.3.20 Changed attribute for cNvSpPr element (Part 1, §20.5.2.9)

Attributes	Description
txBox (Text Box)	Specifies that the corresponding shape is a text box and thus should be treated as such by the generating application. If this attribute is omitted then it is assumed that the

Attributes	Description
Namespace: …/drawingml/2006/main	corresponding shape is not specifically a text box. [*Note*: Because a shape is not specified to be a text box does not mean that it cannot have text attached to it. A text box is merely a specialized shape with specific properties. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

17.3.21 Changed attribute for contentPart element (Part 1, §20.5.2.12)

Attributes	Description
id (Relationship to Part) Namespace: …/officeDocument/2006/relationships	Specifies the relationship ID to a content part. [*Example*: Consider an XML element that has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

17.3.22 Changed attribute for ext element (Part 1, §20.5.2.14)

Attributes	Description
cx (Extent Width) Namespace: …/drawingml/2006/main	Specifies the width of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `<… cx="1828800" cy="200000"/>` The cx attribute specifies that this object has a width of `1828800` EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (Part 1, §20.1.10.42).

Attributes	Description
cy (Extent Height) Namespace: …/drawingml/2006/main	Specifies the height of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `< … cx="1828800" cy="200000"/>` The cy attribute specifies that this object has a height of `200000` EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (Part 1, §20.1.10.42).

17.3.23 Changed attribute for grpSpPr element (Part 1, §20.5.2.18)

Attributes	Description
bwMode (Black and White Mode) Namespace: …/drawingml/2006/main	Specifies that the group shape should be rendered using only black and white coloring. That is the coloring information for the group shape should be converted to either black or white when rendering the corresponding shapes. No gray is to be used in rendering this image, only stark black and stark white. [*Note*: This does not mean that the group shapes themselves are stored with only black and white color information. This attribute instead sets the rendering mode that the shapes use when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (Part 1, §20.1.10.10).

17.3.24 Changed attribute for pos element (Part 1, §20.5.2.26)

Attributes	Description
x (X-Axis Coordinate) Namespace: …/drawingml/2006/main	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The x attribute defines an x-coordinate of `0`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (Part 1, §20.1.10.16).

Attributes	Description
y (Y-Axis Coordinate) Namespace: .../drawingml/2006/main	Specifies a coordinate on the y-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The y attribute defines a y-coordinate of `100`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (Part 1, §20.1.10.16).

17.3.25 Changed attribute for spPr element (Part 1, §20.5.2.30)

Attributes	Description
bwMode (Black and White Mode) Namespace: .../drawingml/2006/main	Specifies that the picture should be rendered using only black and white coloring. That is the coloring information for the picture should be converted to either black or white when rendering the picture. No gray is to be used in rendering this image, only stark black and stark white. [*Note*: This does not mean that the picture itself that is stored within the file is necessarily a black and white picture. This attribute instead sets the rendering mode that the picture has applied to when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (Part 1, §20.1.10.10).

17.3.26 Changed attribute for xfrm element (Part 1, §20.5.2.36)

Attributes	Description
flipH (Horizontal Flip) Namespace: .../drawingml/2006/main	Specifies a horizontal flip. When true, this attribute defines that the shape is flipped horizontally about the center of its bounding box. [*Example*: The following illustrates the effect of a horizontal flip.  *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

Attributes	Description
flipV (Vertical Flip) Namespace: …/drawingml/2006/main	Specifies a vertical flip. When true, this attribute defines that the group is flipped vertically about the center of its bounding box. [*Example*: The following illustrates the effect of a vertical flip.  *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rot (Rotation) Namespace: …/drawingml/2006/main	Specifies the rotation of the Graphic Frame. The units for that this attribute is specified in reside within the simple type definition referenced below. The possible values for this attribute are defined by the ST_Angle simple type (Part 1, §20.1.10.3).

18. DrawingML - Components Reference Material

18.1 DrawingML - Charts

18.1.1 Table of Contents

This subclause is informative.

End of informative text.

18.1.2 Elements

18.1.2.1 legacyDrawingHF (Legacy Drawing for Headers and Footers)

This element specifies the VML Drawing part that contains any pictures used in the header or footer of the chart.

Attributes	Description
id (Relationship Reference) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID for the relationship for this Chart or Chart Drawing part. The type of relationship needed is specified by the parent element. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RelId) is located in §A.5.1. *end note*]

18.1.3 Simple Types

18.1.3.1 Additional member types for union in ST_DepthPercent

The value space of the following additional member types can be used within the context of this simple type for a document of a transitional conformance class.

- The ST_DepthPercentUShort simple type (§13.1.3.2).

18.1.3.2 ST_DepthPercentUShort (Depth Percent UnsignedShort) (Part 1, §21.2.3.9)

This simple type specifies that its contents contain a whole number between 20 and 2000, whose contents are a percentage. This simple type's contents are a restriction of the W3C XML Schema unsignedShort datatype.

This simple type also specifies the following restrictions:

This simple type has a minimum value of greater than or equal to 20.
This simple type has a maximum value of less than or equal to 2000.

18.1.3.3 Additional member types for union in ST_HPercent (Part 1, §21.2.3.19)

The value space of the following additional member types can be used within the context of this simple type for a document of a transitional conformance class.

- The ST_HPercentUShort simple type (§13.1.3.4).

18.1.3.4 ST_HPercentUShort (Depth Percent UnsignedShort)

This simple type specifies that its contents contain a whole number between 5 and 500, whose contents are a percentage.

This simple type's contents are a restriction of the W3C XML Schema unsignedShort datatype.

This simple type also specifies the following restrictions:

This simple type has a minimum value of greater than or equal to 5.
This simple type has a maximum value of less than or equal to 500.

18.1.3.5 Additional member types for union in ST_GapAmount (Part 1, §21.2.3.16)

The value space of the following additional member types can be used within the context of this simple type for a document of a transitional conformance class.

- The ST_GapAmountUShort simple type (§13.1.3.6).

18.1.3.6 ST_GapAmountUShort (Gap Amount UnsignedShort)

This simple type specifies that its contents contain a whole number between 0 and 500, whose contents are a percentage.

This simple type's contents are a restriction of the W3C XML Schema unsignedShort datatype.

This simple type also specifies the following restrictions:

This simple type has a minimum value of greater than or equal to 0.
This simple type has a maximum value of less than or equal to 500.

18.1.3.7 Additional member types for union in ST_SecondPieSize (Part 1, §21.2.3.41)

The value space of the following additional member types can be used within the context of this simple type for a document of a transitional conformance class.

- The ST_SecondPieSizeUShort simple type (§13.1.3.10).

18.1.3.8 ST_SecondPieSizeUShort (Second Pie Size UnsignedShort)

This simple type specifies that its contents contain a whole number between 5 and 200, whose contents are a percentage.

This simple type's contents are a restriction of the W3C XML Schema unsignedShort datatype.

This simple type also specifies the following restrictions:

This simple type has a minimum value of greater than or equal to 5.
This simple type has a maximum value of less than or equal to 200.

18.1.3.9 Additional member types for union in ST_HoleSize (Part 1, §21.2.3.18)

The value space of the following additional member types can be used within the context of this simple type for a document of a transitional conformance class.

- The ST_HoleSizeUByte simple type (§13.1.3.12).

18.1.3.10 ST_HoleSizeUByte (Hole Size UnsignedByte)

This simple type specifies that its contents contain a whole number between 10 and 90, whose contents are a percentage.

This simple type's contents are a restriction of the W3C XML Schema unsignedByte datatype.

This simple type also specifies the following restrictions:

This simple type has a minimum value of greater than or equal to 10.
This simple type has a maximum value of less than or equal to 90.

18.1.3.11 Additional member types for union in ST_LblOffset (Part 1, §21.2.3.23)

The value space of the following additional member types can be used within the context of this simple type for a document of a transitional conformance class.

- The ST_LblOffsetUShort simple type (§13.1.3.14).

18.1.3.12 ST_LblOffsetUShort (Label Offset UnsignedShort)

This simple type specifies that its contents contain a whole number between 0 and 1000, whose contents are a percentage.

This simple type's contents are a restriction of the W3C XML Schema unsignedShort datatype.

This simple type also specifies the following restrictions:

This simple type has a minimum value of greater than or equal to 0.
This simple type has a maximum value of less than or equal to 1000.

18.1.3.13 Additional member types for union in ST_Overlap (Part 1, §21.2.3.31)

The value space of the following additional member types can be used within the context of this simple type for a document of a transitional conformance class.

- The ST_OverlapByte simple type (§13.1.3.16).

18.1.3.14 ST_OverlapByte (Overlap Byte)

This simple type specifies that its contents contain a whole number between -100 and 100, whose contents are a percentage.

This simple type's contents are a restriction of the W3C XML Schema byte datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to -100.

This simple type has a maximum value of less than or equal to 100.

18.1.3.15 Additional member types for union in ST_BubbleScale (Part 1, §21.2.3.5)

The value space of the following additional member types can be used within the context of this simple type for a document of a transitional conformance class.

- The ST_BubbleScaleUInt simple type (§13.1.3.18).

18.1.3.16 ST_BubbleScaleUInt (Bubble Scale UnsignedInt)

This simple type specifies that its contents contain a whole number between 0 and 300, whose contents are a percentage.

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

This simple type also specifies the following restrictions:

This simple type has a minimum value of greater than or equal to 0.
This simple type has a maximum value of less than or equal to 300.

18.1.3.17 Additional member types for union in ST_Thickness (Part 1, §21.2.3.206)

The value space of the following additional member types can be used within the context of this simple type for a document of a transitional conformance class.

- The W3C XML Schema unsignedInt datatype.

18.2 Changed attributes

The following attributes, which are defined in subclauses within Part 1, §21, "DrawingML - Components Reference Material", have different source relationships when used in documents of the Transitional conformance class:

18.2.1 Changed attribute for hlinkClick element (Part 1, §21.1.2.3.5)

Attributes	Description
id (Drawing Object Hyperlink Target) Namespace: .../officeDocument/2006/relationships	Specifies the relationship id that when looked up in this slides relationship file contains the target of this hyperlink. This attribute cannot be omitted. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

18.2.2 Changed attribute for hlinkMouseOver element (Part 1, §21.1.2.3.6)

Attributes	Description
id (Drawing Object Hyperlink Target) Namespace: .../officeDocument/2006/relationships	Specifies the relationship id that when looked up in this slides relationship file contains the target of this hyperlink. This attribute cannot be omitted. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

18.2.3 Changed attribute for chart element (Part 1, §21.2.2.26)

Attributes	Description
id (Relationship Reference) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID for the relationship for this Chart or Chart Drawing part. The type of relationship needed is specified by the parent element. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

18.2.4 Changed attribute for clrMapOvr element (Part 1, §21.2.2.30)

Attributes	Description
accent1 (Accent 1) Namespace: .../drawingml/2006/main	Specifies a color defined that is associated as the accent 1 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).

Attributes	Description
accent2 (Accent 2) Namespace: …/drawingml/2006/main	Specifies a color defined that is associated as the accent 2 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).
accent3 (Accent 3) Namespace: …/drawingml/2006/main	Specifies a color defined that is associated as the accent 3 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).
accent4 (Accent 4) Namespace: …/drawingml/2006/main	Specifies a color defined that is associated as the accent 4 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).
accent5 (Accent 5) Namespace: …/drawingml/2006/main	Specifies a color defined that is associated as the accent 5 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).
accent6 (Accent 6) Namespace: …/drawingml/2006/main	Specifies a color defined that is associated as the accent 6 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).
bg1 (Background 1) Namespace: …/drawingml/2006/main	A color defined that is associated as the first background color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).
bg2 (Background 2) Namespace: …/drawingml/2006/main	Specifies a color defined that is associated as the second background color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).
folHlink (Followed Hyperlink) Namespace: …/drawingml/2006/main	Specifies a color defined that is associated as the color for a followed hyperlink. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).

Attributes	Description
hlink (Hyperlink) Namespace: .../drawingml/2006/main	Specifies a color defined that is associated as the color for a hyperlink. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).
tx1 (Text 1) Namespace: .../drawingml/2006/main	Specifies a color defined that is associated as the first text color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).
tx2 (Text 2) Namespace: .../drawingml/2006/main	Specifies a color defined that is associated as the second text color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (Part 1, §20.1.10.14).

18.2.5 Changed attribute for externalData element (Part 1, §21.2.2.63)

Attributes	Description
id (Relationship Reference) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID for the relationship for this chart. The relationship explicitly targeted by this attribute shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/package . The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

18.2.6 Changed attribute for spPr element (Part 1, §21.2.2.197)

Attributes	Description
bwMode (Black and White Mode) Namespace: .../drawingml/2006/main	Specifies that the picture should be rendered using only black and white coloring. That is the coloring information for the picture should be converted to either black or white when rendering the picture. No gray is to be used in rendering this image, only stark black and stark white. [*Note*: This does not mean that the picture itself that is stored within the file is necessarily a black and white picture. This attribute instead sets the rendering mode that the picture has applied to when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (Part 1, §20.1.10.10).

18.2.7 Changed attribute for userShapes element (Part 1, §21.2.2.221)

Attributes	Description
id (Relationship Reference) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID for the relationship for this Chart or Chart Drawing part. The type of relationship needed is specified by the parent element. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

18.2.8 Changed attribute for blipFill element (Part 1, §21.3.2.2)

Attributes	Description
dpi (DPI Setting) Namespace: .../drawingml/2006/main	Specifies the DPI (dots per inch) used to calculate the size of the blip. If not present or zero, the DPI in the blip is used. [*Note*: This attribute is primarily used to keep track of the picture quality within a document. There are different levels of quality needed for print than on-screen viewing and thus a need to track this information. *end note*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
rotWithShape (Rotate With Shape) Namespace: .../drawingml/2006/main	Specifies that the fill should rotate with the shape. That is, when the shape that has been filled with a picture and the containing shape (say a rectangle) is transformed with a rotation then the fill is transformed with the same rotation. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

18.2.9 Changed attribute for cNvPicPr element (Part 1, §21.3.2.6)

Attributes	Description
preferRelativeResize (Relative Resize Preferred) Namespace: .../drawingml/2006/main	Specifies if the user interface should show the resizing of the picture based on the picture's current size or its original size. If this attribute is set to true, then scaling is relative to the original picture size as opposed to the current picture size. [*Example*: Consider the case where a picture has been resized within a document and is now 50% of the originally inserted picture size. Now if the user chooses to make a later adjustment to the size of this picture within the generating application, then the value of this attribute should be checked. If this attribute is set to true then a value of 50% is shown. Similarly, if this attribute is set to false, then a value of 100% should be shown because the picture has not yet been resized from its current (smaller) size. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean

Attributes	Description
	datatype.

18.2.10 Changed attribute for cNvPr element (Part 1, §21.3.2.7)

Attributes	Description
descr (Alternative Text for Object) Namespace: …/drawingml/2006/main	Specifies alternative text for the current DrawingML object, for use by assistive technologies or applications that do not display the current object. If this element is omitted, then no alternative text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: `<… descr="A picture of a bowl of fruit">` The descr attribute contains alternative text that can be used in place of the actual DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
hidden (Hidden) Namespace: …/drawingml/2006/main	Specifies whether this DrawingML object is displayed. When a DrawingML object is displayed within a document, that object can be hidden (i.e., present, but not visible). This attribute determines whether the object is rendered or made hidden. [*Note*: An application can have settings that allow this object to be viewed. *end note*] If this attribute is omitted, then the parent DrawingML object shall be displayed (i.e., not hidden). [*Example*: Consider an inline DrawingML object that must be hidden within the document's content. This setting would be specified as follows: `<… hidden="true" />` The hidden attribute has a value of `true`, which specifies that the DrawingML object is hidden and not displayed when the document is displayed. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Unique Identifier) Namespace: …/drawingml/2006/main	Specifies a unique identifier for the current DrawingML object within the current document. This ID can be used to assist in uniquely identifying this object so that it can be referred to by other parts of the document. If multiple objects within the same document share the same id attribute value, then the document shall be considered non-conformant. [*Example*: Consider a DrawingML object defined as follows:

Attributes	Description
	`<… id="10" … >` The id attribute has a value of `10`, which is the unique identifier for this DrawingML object. *end example*] The possible values for this attribute are defined by the ST_DrawingElementId simple type (Part 1, §20.1.10.21).
name (Name) Namespace: …/drawingml/2006/main	Specifies the name of the object. [*Note*: Typically, this is used to store the original file name of a picture object. *end note*] [*Example*: Consider a DrawingML object defined as follows: `< … name="foo.jpg" >` The name attribute has a value of `foo.jpg`, which is the name of this DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
title (Title) Namespace: …/drawingml/2006/main	Specifies the title (caption) of the current DrawingML object. If this attribute is omitted, then no title text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: `<… title="Process Flow Diagram">` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

18.2.11 Changed attribute for cNvSpPr element (Part 1, §21.3.2.8)

Attributes	Description
txBox (Text Box) Namespace: …/drawingml/2006/main	Specifies that the corresponding shape is a text box and thus should be treated as such by the generating application. If this attribute is omitted then it is assumed that the corresponding shape is not specifically a text box. [*Note*: Because a shape is not specified to be a text box does not mean that it cannot have text attached to it. A text box is merely a specialized shape with specific properties. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

18.2.12 Changed attribute for ext element (Part 1, §21.3.2.10)

Attributes	Description
cx (Extent Width) Namespace: …/drawingml/2006/main	Specifies the width of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `<… cx="1828800" cy="200000"/>` The cx attribute specifies that this object has a width of 1828800 EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (Part 1, §20.1.10.42).
cy (Extent Height) Namespace: …/drawingml/2006/main	Specifies the height of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `< … cx="1828800" cy="200000"/>` The cy attribute specifies that this object has a height of 200000 EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (Part 1, §20.1.10.42).

18.2.13 Changed attribute for grpSpPr element (Part 1, §21.3.2.14)

Attributes	Description
bwMode (Black and White Mode) Namespace: .../drawingml/2006/main	Specifies that the group shape should be rendered using only black and white coloring. That is the coloring information for the group shape should be converted to either black or white when rendering the corresponding shapes. No gray is to be used in rendering this image, only stark black and stark white. [*Note*: This does not mean that the group shapes themselves are stored with only black and white color information. This attribute instead sets the rendering mode that the shapes use when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (Part 1, §20.1.10.10).

18.2.14 Changed attribute for spPr element (Part 1, §21.3.2.23)

Attributes	Description
bwMode (Black and White Mode) Namespace: .../drawingml/2006/main	Specifies that the picture should be rendered using only black and white coloring. That is the coloring information for the picture should be converted to either black or white when rendering the picture. No gray is to be used in rendering this image, only stark black and stark white. [*Note*: This does not mean that the picture itself that is stored within the file is necessarily a black and white picture. This attribute instead sets the rendering mode that the picture has applied to when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (Part 1, §20.1.10.10).

18.2.15 Changed attribute for xfrm element (Part 1, §21.3.2.28)

Attributes	Description
flipH (Horizontal Flip) Namespace: …/drawingml/2006/main	Specifies a horizontal flip. When true, this attribute defines that the shape is flipped horizontally about the center of its bounding box. [*Example*: The following illustrates the effect of a horizontal flip.  *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
flipV (Vertical Flip) Namespace: …/drawingml/2006/main	Specifies a vertical flip. When true, this attribute defines that the group is flipped vertically about the center of its bounding box. [*Example*: The following illustrates the effect of a vertical flip.  *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rot (Rotation) Namespace: …/drawingml/2006/main	Specifies the rotation of the Graphic Frame. The units for which this attribute is specified in reside within the simple type definition referenced below. The possible values for this attribute are defined by the ST_Angle simple type (Part 1, §20.1.10.3).

18.2.16 Changed attribute for relIds element (Part 1, §21.4.2.22)

Attributes	Description
cs (Explicit Relationship to Diagram Colors Part) Namespace: …/officeDocument	Specifies the relationship ID for the explicit relationship to the Diagram Colors part used by this diagram. This relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/diagramColors or the document shall be considered non-conformant.

Attributes	Description
/2006/relationships	The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).
dm (Explicit Relationship to Diagram Data Part) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID for the explicit relationship to the Diagram Data part used by this diagram. This relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/diagramData or the document shall be considered non-conformant. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).
lo (Explicit Relationship to Diagram Layout Definition Part) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID for the explicit relationship to the Diagram Layout Definition part used by this diagram. This relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/diagramLayout or the document shall be considered non-conformant. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).
qs (Explicit Relationship to Style Definition Part) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID for the explicit relationship to the Diagram Style part used by this diagram. This relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/diagramQuickStyle or the document shall be considered non-conformant. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

18.2.17 Changed attribute for shape element (Part 1, §21.4.2.27)

Attributes	Description
blip (Relationship to Image Part) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID of the explicit relationship to an image that shall be used as the image for the contents of this shape. This relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/image or the document shall be considered non-conformant. The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

18.2.18 Changed attribute for spPr element (Part 1, §21.4.3.7)

Attributes	Description
bwMode (Black and White Mode) Namespace: .../drawingml/2006/main	Specifies that the picture should be rendered using only black and white coloring. That is the coloring information for the picture should be converted to either black or white when rendering the picture. No gray is to be used in rendering this image, only stark black and stark white. [*Note*: This does not mean that the picture itself that is stored within the file is necessarily a black and white picture. This attribute instead sets the rendering mode that the picture has applied to when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (Part 1, §20.1.10.10).

18.2.19 Changed attribute for sp3d element (Part 1, §21.4.5.6)

Attributes	Description

contourW (Contour Width)
 	
Namespace: .../drawingml/2006/main
	Defines the width of the contour on the shape.
	
	[*Example*: Consider the following example of a contourW in use within the sp3d element:
	
	`<a:sp3d extrusionH="165100" contourW="50800"`
	`  prstMaterial="plastic">`
	`  <a:bevelT w="254000" h="254000"/>`
	`  <a:bevelB w="254000" h="254000"/>`
	`  <a:extrusionClr>`
	`  <a:srgbClr val="FF0000"/>`
	`  </a:extrusionClr>`
	`  <a:contourClr>`
	`  <a:schemeClr val="accent3"/>`
	`  </a:contourClr>`
	`</a:sp3d>`
	
	In this example, we see a countourW defined as `50800`. *end example*]
	
	The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (Part 1, §20.1.10.42).

Attributes	Description

extrusionH
(Extrusion Height)
 	
Namespace:
…/drawingml/2006/main
	Defines the height of the extrusion applied to the shape.
	
	[*Example*: Consider the following example of an extrusionH in use within the sp3d element:
	
	`<a:sp3d extrusionH="165100" contourW="50800" prstMaterial="plastic">`
	`  <a:bevelT w="254000" h="254000"/>`
	`  <a:bevelB w="254000" h="254000"/>`
	`  <a:extrusionClr>`
	`  <a:srgbClr val="FF0000"/>`
	`  </a:extrusionClr>`
	`  <a:contourClr>`
	`  <a:schemeClr val="accent3"/>`
	`  </a:contourClr>`
	`</a:sp3d>`
	
	In this example, we see a extrusionH defined as `165100`. *end example*]
	
	The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (Part 1, §20.1.10.42).

prstMaterial
(Preset Material Type)
 	
Namespace:
…/drawingml/2006/main
	Defines the preset material that is combined with the lighting properties to give the final look and feel of a shape.
	
	[*Example*: Consider the following example of a prstMaterial in use within the sp3d element:
	
	`<a:sp3d extrusionH="165100" contourW="50800" prstMaterial="plastic">`
	`  <a:bevelT w="254000" h="254000"/>`
	`  <a:bevelB w="254000" h="254000"/>`
	`  <a:extrusionClr>`
	`  <a:srgbClr val="FF0000"/>`
	`  </a:extrusionClr>`
	`  <a:contourClr>`
	`  <a:schemeClr val="accent3"/>`
	`  </a:contourClr>`
	`</a:sp3d>`
	
	In this example, we see a prstMaterial defined as `plastic`. *end example*]
	
	The possible values for this attribute are defined by the ST_PresetMaterialType simple type (Part 1, §20.1.10.50).

Attributes	Description
z (Shape Depth) Namespace: .../drawingml/2006/main	Defines the z coordinate for the 3D shape. The possible values for this attribute are defined by the ST_Coordinate simple type (Part 1, §20.1.10.16).

19. VML Reference Material

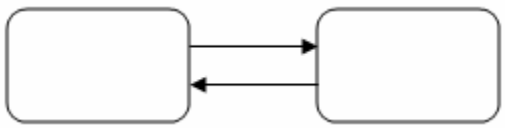
[No documentation has been entered for this section heading.]

19.1 VML

VML is a language for defining graphical objects in cases where DrawingML does not apply, such as text boxes and shapes in WordprocessingML documents and comments and controls in SpreadsheetML documents. The urn:schemas-microsoft-com:vml namespace provides the base elements and attributes for defining shape primitives. The urn:schemas-microsoft-com:office:office, urn:schemas-microsoft-com:office:word, urn:schemas-microsoft-com:office:excel and urn:schemas-microsoft-com:office:powerpoint namespaces define elements that layer on information beyond the baseline graphical definition. To maintain backward compatibility, all VML namespaces defined in ISO/IEC 29500 maintain the legacy namespace structure used by the existing corpus of binary documents.

[*Note*: The VML format is a legacy format used in an existing corpus of binary documents and is included and fully defined in ISO/IEC 29500 for backwards compatibility reasons. The DrawingML format is a newer and richer format created with the goal of eventually replacing any uses of VML in the Office Open XML formats. VML should be considered a transitional format included in Office Open XML for legacy reasons only and new applications that need a file format for drawings are strongly encouraged to use preferentially DrawingML .*end note*]

[*Example*: Assume the following shapes exist in a WordprocessingML document:



The drawing consists of four shapes. The arrows are specified by extending the shape type base definition in the shapetype element. Each shape representing an arrow references the shapetype it is extending via its type attribute.

```
<v:shapetype id="_x0000_t32" coordsize="21600,21600" o:spt="32" o:oned="t"
  path="m,l21600,21600e" filled="f">
  <v:path arrowok="t" fillok="f" o:connecttype="none"/>
  <o:lock v:ext="edit" shapetype="t"/>
</v:shapetype>

<v:shape id="_x0000_s1030" type="#_x0000_t32" style="position:absolute;left:0;
  text-align:left;margin-left:105pt;margin-top:36pt;width:48pt;height:0;flip:x;
  z-index:251661312" o:connectortype="straight">
```



```
    <v:stroke endarrow="block"/>
  </v:shape>

  <v:shape id="_x0000_s1029" type="#_x0000_t32" style="position:absolute;left:0;
    text-align:left;margin-left:105pt;margin-top:21.75pt;width:48pt;height:0;
    z-index:251660288" o:connectortype="straight">
    <v:stroke endarrow="block"/>
  </v:shape>
```

The rounded rectangles use the VML roundrect element.

```
  <v:roundrect id="_x0000_s1028" style="position:absolute;left:0;
    text-align:left;margin-left:153pt;margin-top:8.25pt;width:68.25pt;height:42pt;
    z-index:251659264" arcsize="10923f"/>

  <v:roundrect id="_x0000_s1027" style="position:absolute;left:0;
    text-align:left;margin-left:36.75pt;margin-top:8.25pt;width:68.25pt;
    height:42pt;z-index:251658240" arcsize="10923f"/>
```

end example]

Throughout VML, numeric values that are allowed to take units can be specified in: cm (centimeters), mm (millimeters), in (inches), pt (points), pc (picas), px (pixels).

19.1.1 Table of Contents

This subclause is informative.

End of informative text.

19.1.2 Elements

The following elements comprise the contents of the urn:schemas-microsoft-com:vml namespace:

[*Note*: As the VML format is a format provided for backward compatibility, many VML elements are defined in the same urn:schemas-microsoft-com:vml namespace that is already used by millions of documents already using VML. *end note*]

19.1.2.1 arc (Arc Segment)

This element specifies an arc defined as a segment of an oval. The CSS2 style content width and height define the width and height of that oval. The arc is defined by the intersection of the oval with the start and end radius vectors given by the angles. The angles are calculated on the basis of a circle (width equal to height) which is then scaled anisotropically to the desired width and height.

[*Example*: The following specifies a simple half-circle arc open at the top:

```
<v:arc
  style="position:relative;top:120;left:20;width:200;height:200"
  startangle="90" endangle="270">
</v:arc>
```

The shape looks like this:

end example]

Attributes	Description

allowincell (Allow in Table Cell)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can be placed in a table. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:allowincell="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

allowoverlap (Allow Shape Overlap)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can overlap another shape. Default is `true`. If `false`, the shape shifts left or right so as not to overlap another shape, similar to the behavior of the HTML float attribute.
	
	[*Example*:
	
	`<v:shape … o:allowoverlap="false" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

alt (Alternate Text)	Specifies alternative text describing the graphical object. This text should provide a brief description of the shape for use by accessibility tools. Default is no value.
	
	[*Example*: The alt text describes the basic shape:
	
	`<v:shape … fillcolor="red"`
	`  alt="Red rectangle">`
	`</v:shape>`
	
	The alt text describes the contents of a shape displaying an image:
	
	`<v:shape … alt="Picture of a sunset">`
	`</v:shape>`

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderbottomcolor (Bottom Border Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the bottom border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderbottomcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderleftcolor (Border Left Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the left border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderleftcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderrightcolor (Border Right Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the right border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderrightcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

bordertopcolor (Border Top Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the top border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:bordertopcolor="red" … >`
	`</v:shape>`

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema string datatype.

bullet (Graphical Bullet)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the shape is a graphical bullet. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:bullet="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

button (Button Behavior Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape exhibits button press behavior on click. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:button="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

bwmode (Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies how a shape renders for black-and-white output devices. When a shape is printed on a black-and-white printer or displayed in a black-and-white view in an application, several options are possible. Default is `auto`, which uses o:bwnormal for normal black-and-white rendering and o:bwpure for pure black-and-white rendering.
	
	bwnormal and bwpure are subordinate to bwmode. If bwmode is "auto" then the value for bwnormal or bwpure is used depending on what the output format is. An application can define for itself what, if any, difference there is between normal B&W and pure B&W. [*Example*: Normal B&W might allow greyscale and pure B&W might not. *end example*]
	[*Example*: This shape renders in grayscale in a black-and-white environment:
	
	`<v:shape … o:bwmode="grayscale" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

Attributes	Description

bwnormal (Normal Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the black-and-white mode for normal black-and-white output devices. Default is `auto`.
	
	[*Example*: This shape renders in a pale grayscale in a normal black-and-white environment:
	
	`<v:shape … o:bwmode="auto" o:bwnormal="lightgrayscale" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

bwpure (Pure Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the black-and-white mode for pure black-and-white output devices. Default is `auto`.
	
	[*Example*: This shape renders in high contrast when in a pure black-and-white environment:
	
	`<v:shape … o:bwmode="auto" o:bwpure="highcontrast" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

chromakey (Image Transparency Color)	Specifies a color value that is transparent and show anything behind the shape. Default is no value.
	
	[*Example*:
	
	`<v:image … chromakey="white" …>`
	`</v:image>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

class (CSS Reference)	Specifies a reference to the definition of a CSS style. Default is no value.
	
	[*Example*: The snippets below are equivalent:
	
	`…`
	`.narrowstyle {width:50;height:100}`
	`…`

Attributes	Description

	`<v:shape … class="narrowstyle"`
	`  style="top:1;left:1">`
	`</v:shape>`
	
	`<v:shape … style="top:1;left:1;`
	`  width:50;height:100">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

clip (Clipping Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the clipping region is active. This is used in conjunction with the clippath (§19.2.2.3) element to create a clipping region.
	
	[*Example*:
	
	`<v:shape … o:clip="true">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

cliptowrap (Clip to Wrapping Polygon)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the clipping region for the shape aligns with the wrapping polygon that tightly surrounds the entire shape (essentially, that the shape shall not be drawn beyond its wrapping polygon's extents – if it does, the shape shall be clipped). Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:cliptowrap="true">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

connectortype (Shape Connector Type)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of connector used for joining shapes. Default is `straight`.
	
	[*Example*:
	
	`<v:shape … o:connectortype="elbow" … >`
	`</v:shape>`

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_ConnectorType simple type (§19.2.3.7).

coordorigin (Coordinate Space Origin)	Specifies the coordinate of the top left corner of the shape's coordinate space. This determines the position of the (0,0) coordinate space origin within the shape's bounding box. Default is "0,0", which places the (0,0) origin at the top left corner of the bounding box.
	
	This affects shape properties that specify coordinate positions, such as the path attribute. Thus a path can be defined against a generic (0,0) origin and the coordorigin value translates the entire path within the shape's bounding space.
	
	[*Example*: The horizontal and vertical coordinate space ranges from -100 to +100 because the coordinate space (coordsize) is 200 by 200 and the top left coordinate is (-100,-100). The (0,0) origin lies at the center of the shape's bounding box, as evidenced by the position of the shape's path within the coordinate space:
	
	`<v:shape … coordsize="200,200"`
	`  coordorigin="-100,-100"`
	`  path="m 0,0 l 0,50, 50,50, 50,0 x e">`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

coordsize (Coordinate Space Size)	Specifies the size of the shape's coordinate space in coordinate units. Default is "1000,1000".
	
	The physical size of a coordinate unit length is determined by both the size of the coordinate space (coordsize) and the size of the shape (style width and height). The coordsize attribute defines the number of horizontal and vertical subdivisions into which the shape's bounding box is divided. The combination of coordsize and style width/height effective scales the shape anisotropically.
	
	[*Example*: The path is 50 units wide and tall, which is 25% of the size of the coordinate space:
	
	`<v:shape … coordsize="200,200"`
	`  coordorigin="-100,-100"`
	`  path="m 0,0 l 0,50, 50,50, 50,0 x e">`

Attributes	Description
	`</v:shape>` ❑ *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

dgmlayout (Diagram Node Layout Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of automatic layout to apply to the diagram node. This is only meaningful if the shape is a node in an organization chart, which is denoted by the `orgchart` value of the editas attribute of the group element.
	[*Example*:
	
	`<v:shape … dgmlayout="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).

dgmlayoutmru (Diagram Node Recent Layout Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of automatic layout most recently used on the child elements of the diagram node. This is only meaningful if the shape is a node in anorganization chart, which is denoted by the `orgchart` value of the editas attribute of the group element.
	
	[*Example*:
	
	`<v:shape … dgmlayout="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).

dgmnodekind (Diagram Node Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies an optional, application-defined parameter that is intended to be used by the application to tag different types of nodes in a diagram.
	
	[*Example*:
	
	`<v:shape … dgmnodekind="1">`
	`</v:shape>`
	
	end example]

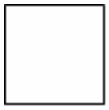
Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

doubleclicknotify (Double-click Notification Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that an event message is sent when a shape is double-clicked. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:doubleclicknotify="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

endAngle (Ending Angle)	Specifies the angle that defines the endpoint of the arc. The angle is measured in degrees clockwise from the vertical. Default is `90`.
	
	[*Example*: This arc ends at the bottom center of the shape's region:
	
	`<v:arc … endangle="180" … >`
	`</v:arc>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema decimal datatype.

fillcolor (Fill Color)	Specifies the color to use for the fill. Default is `white`. If the fill element (§19.1.2.5) is present, its color attribute takes precedence. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*: This shape is red if its fill is visible:
	
	`<v:shape … fillcolor="red" … >`
	`</v:shape>`
	
	This is equivalent to:
	
	`<v:shape … fillcolor="#ff0000" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type

Attributes	Description
	(§20.1.2.3).

filled (Shape Fill Toggle)	Specifies whether the closed path is filled. Default is `true`. This attribute is overridden by the fill on attribute.
	
	[*Example*:
	
	`<v:shape … filled="f"`
	`  fillcolor="red" …>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

forcedash (Force Dashed Outline)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a dashed outline is used to draw a shape when a shape has no line or fill. Default is `false`.
	
	Used by PresentationML placeholders to draw a dashed outline when there is no line and no fill for a shape.
	
	[*Example*:
	
	`<v:shape … o:forcedash="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hr (Horizontal Rule Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that a shape is a horizontal rule. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hr="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hralign (Horizontal	Specifies the alignment of a horizontal rule. Default is `left`.

Attributes	Description

Rule Alignment)
 	
Namespace: urn:schemas-microsoft-com:office:office
	[*Example*:
	
	`<v:shape … o:hralign="center" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_HrAlign simple type (§19.2.3.16).

href (Hyperlink Target)	Specifies a hyperlink URL target for the shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … href="http://www.openxmlformats.org" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

hrnoshade (Horizontal Rule 3D Shading Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the horizontal rule does not have 3-D shading. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hrnoshade="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hrpct (Horizontal Rule Length Percentage)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the length of a horizontal rule as a percentage of page width. Default is 0.
	
	[*Example*:
	
	`<v:shape … o:hrpct="85" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema float datatype.

hrstd (Horizontal Rule Standard Display Toggle)	Specifies whether a shape is displayed as a standard horizontal rule. Only applies if hr is `true`. Default is `false`.

Attributes	Description

Namespace: urn:schemas-microsoft-com:office:office	[*Example*:
	
	`<v:shape … o:hrstd="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

id (Unique Identifier)	Specifies a unique identifier that can be used to reference a VML object.
	
	Default is no value.
	
	[*Example*:
	
	`<v:shape … id="myShape" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

insetmode (Text Inset Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the application calculates the internal text margin instead of using the inset attribute. Default is `custom`. This attribute is only meaningful for text boxes.
	
	[*Example*:
	
	`<v:shape … o:insetmode="auto" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_InsetMode simple type (§19.2.3.17).

insetpen (Inset Border From Path)	Specifies that the border shall be displayed inside of the path defining the shape, rather than along the path (the default border placement), or outside of the path as might be done with an image.
	
	[*Example*:
	
	`<v:shape … insetpen="true" … >`
	`</v:shape>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

ole (Embedded Object Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the shape is an embedded object. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:ole="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

oleicon (Embedded Object Icon Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether an embedded object is displayed as an icon. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oleicon="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

oned (Shape Handle Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the extra handles of a shape are hidden. If `true`, hides all shape handles except the top left and bottom right; that is, the same handles that are used for a straight line segment. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oned="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

opacity (Fill Color Opacity)	Specifies the opacity of the primary fill color. Default is 1.0. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. [*Example*: A value of "52429f" represents 52429/65536 or 0.8. *end example*]
	
	[*Example*: The red color is 25% opaque:
	
	`<v:fill type="gradient" color="red"`
	`color2="blue" opacity=".25">`

Attributes	Description
	`</v:fill>`  `opacity="1"`  `opacity=".25"` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

preferrelative (Relative Resize Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the original size of an object is saved after reformatting. Default is `false`. If `true`, the original size of the object is stored and all resizing is based on a percentage of that original size. Otherwise, each resizing resets the scale to 100%.
	
	[*Example*:
	
	`<v:shape … o:preferrelative="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

print (Print Toggle)	Specifies whether the shape is printed. Default is `true`.
	
	[*Example*:
	
	`<v:shape … print="false" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

regroupid (Regroup ID)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies a previous group for a shape. An ID number is used to identify groups of shapes that are no longer grouped. This allows shapes to be regrouped programmatically.
	
	[*Example*: The shape was part of a group identified by the ID 040754:
	
	`<v:shape … o:regroupid="040754" … >`
	`</v:shape>`

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema integer datatype.
spid (Optional String) Namespace: urn:schemas-microsoft-com:office:office	Specifies an optional string that an application can use to Identify the particular shape. Default is no value. The possible values for this attribute are defined by the W3C XML Schema string datatype.
spt (Optional Number) Namespace: urn:schemas-microsoft-com:office:office	Specifies an optional number that an application can use to associate the particular shape with a defined shape type. Default is 0. The possible values for this attribute are defined by the W3C XML Schema float datatype.

startAngle (Starting Angle)	Specifies an angle that defines the starting point of the arc. The angle is measured in degrees clockwise from the vertical.
	
	Default is `0`.
	
	[*Example*: This arc begins in the upper-right quadrant:
	
	`<v:arc … startangle="45" … >`
	`</v:arc>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema decimal datatype.

strokecolor (Shape Stroke Color)	Specifies the primary color of the brush to use to stroke the path of the shape. Default is `black`. The color attribute of the stroke element (§19.1.2.21) overrides this. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*:
	
	`<v:shape … strokecolor="red" …>`
	`</v:shape>`

Attributes	Description
	 *end example*] The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

stroked (Shape Stroke Toggle)	Specifies whether the path defining the shape is stroked with a solid line. The stroke element (§19.1.2.21) defines other strokes. The on attribute of the stroke element overrides this attribute. Default is `true`.
	
	[*Example*:
	
	`<v:shape … fillcolor="red"`
	`  stroked="false" strokecolor="blue"…>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

strokeweight (Shape Stroke Weight)	Specifies the width of the brush to use to stroke the path. Default is 1 point. If a number is given without units, the emu is used. The weight attribute of the stroke element (§19.1.2.21) overrides this attribute.
	
	[*Example*:
	
	`<v:shape … strokeweight="3pt" … >`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

style (Shape Styling Properties)	Specifies the CSS2 styling properties of the shape. The CSS2 (Cascading Style Sheets, Level 2) specification, a Recommendation of the World Wide Web Consortium, is available

Attributes	Description

	here: http://www.w3.org/TR/REC-CSS2.
	
	This attribute uses a semi-colon delimited list of “name:value” pairs, the syntax defined by section 4.1.8 of the CSS2 specification for grouped declarations (without the surrounding braces). The following tables define the allowed properties and the VML treatment of each.
	
	[*Example*:
	`<v:shape … style=’position:absolute;width:100pt;height:50pt’ …`
	`>`
	`</v:shape>`
	end example]

Property	Description
`flip`	Specifies that the orientation of a shape is flipped. Default is no value. Allowed values are: • x - Flip along the y-axis, reversing the x-coordinates. • y - Flip along the x-axis, reversing the y-coordinates. • xy - Flip along both the y- and x-axis. • yx - Flip along both the x- and y-axis.
`height`	Specifies the height of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
`left`	Specifies the position of the left of the containing block of the shape relative to the element left of it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the

Attributes	Description	
		parent object's width.
	`margin-bottom`	Specifies the position of the bottom of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`margin-left`	Specifies the position of the left of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-right`	Specifies the position of the right of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-top`	Specifies the position of the top of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page.

Attributes	Description	
		• <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`mso-position-horizontal`	Specifies the horizontal positioning data for objects in WordprocessingML documents. Default is `absolute`. Allowed values are: • `absolute` • `left` • `center` • `right` • `inside` • `outside`
	`mso-position-horizontal-relative`	Specifies relative horizontal position data for objects in WordprocessingML documents. This modifies the `mso-position-horizontal` property. Default is `text`. Allowed values are: • `margin` • `page` • `text` • `char`
	`mso-position-vertical`	Specifies the vertical positioning data for objects in WordprocessingML documents. Default is `absolute`. Allowed values are: • `absolute` • `top` • `center` • `bottom` • `inside` • `outside`
	`mso-position-vertical-relative`	Specifies relative vertical position data for objects in WordprocessingML documents. This modifies the `mso-position-vertical` property. Default is `text`. Allowed values are: • `margin` • `page`

Attributes	Description
	• text • line
mso-wrap-distance-bottom	Specifies the distance from the bottom of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
mso-wrap-distance-left	Specifies the distance from the left side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
mso-wrap-distance-right	Specifies the distance from the right side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
mso-wrap-distance-top	Specifies the distance from the top of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
mso-wrap-edited	Specifies whether the wrap coordinates were customized by the user. If the wrap coordinates are generated by an editor, this property is true; otherwise they were customized by a user. Default is false.
mso-wrap-style	Specifies the wrapping mode for text in shapes in WordprocessingML documents. Default is square. Allowed values are: • square - Wraps text inside the shape in a square. • none - Text does not wrap.
position	Specifies the kind of positioning used to place an element. Default is static. When the element is contained inside a group, this property shall be absolute. Allowed values are: • static - The element is positioned according to the normal flow of the page. The top and left properties are ignored. If the object is anchored inline, this value is used. • absolute - The element is positioned relative to the parent, using the top and left properties. • relative - The element is positioned according to the

Attributes	Description
	normal flow of the page, but the `top` and `left` properties are used. The overlap of overlapping elements is governed by the `z-index` property.
	`rotation` — Specifies the angle that a shape is rotated, in degrees. Default is 0. Positive angles are clockwise.
	`top` — Specifies the position of the top of the containing block of the shape relative to the element above it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`visibility` — Specifies whether a shape is displayed. Only `inherit` and `hidden` are used; any other values are mapped to `inherit`. Default is `inherit`. Allowed values are: • `hidden` - The shape is not visible, but is still part of the flow of the objects in the browser. Mouse events are not processed. • `inherit` - The visibility state is inherited from the parent of the shape.
	`width` — Specifies the width of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`z-index` — Specifies the display order of overlapping shapes. Default is 0. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Uses the order that the shapes appear in the page,

Attributes	Description
	bottom to top. • <order>- A number that represents the stacking precedence. Shapes with higher numbers are placed on top of those with lower numbers. Negative numbers are allowed.

The following properties are only used by the textbox element (§19.1.2.22):

Property	Description
`direction`	Specifies the direction of the text in the textbox. Default is `ltr`. This property is superceded by the `mso-direction-alt` property if that is specified. Allowed values are: • `ltr` - Text is displayed left-to-right. • `rtl` - Text is displayed right-to-left.
`layout-flow`	Determines the flow of the text layout in a textbox. Default is `horizontal`. Allowed values are: • `horizontal` - Text is displayed horizontally. • `vertical` - Text is displayed vertically. • `vertical-ideographic` - Ideographic text is displayed vertically. • `horizontal-ideographic` - Ideographic text is displayed horizontally.
`mso-direction-alt`	Specifies an alternate direction for text in textboxes. Overrides the `direction` property. The only allowed value is `context`.
`mso-fit-shape-to-text`	Specifies whether the shape stretches to fit the text in the textbox. Default is `false`.
`mso-fit-text-to-shape`	Specifies whether the text stretches to fit the textbox. Default is `false`.
`mso-layout-flow-alt`	Specifies the alternate layout flow for text in textboxes. This property is used instead of `layout-flow` when the layout flow is from bottom to top for non-ideographic languages. Its only value is `bottom-to-top`.
`mso-next-textbox`	Specifies the ID of the next textbox in a series. Used to keep track of a set of linked textboxes. Default is no value.

Attributes	Description
	`mso-rotate` — Specifies a specific rotation value for text in a textbox. Default is 0. Allowed values are: • 0 • 90 • 180 • -90
	`mso-text-scale` — Specifies the scaling factor for fitting text to shapes. Default is 0. This property is only used if `mso-fit-text-to-shape` is `true`.
	`v-text-anchor` — Specifies the vertical anchoring of text in a textbox. Default is `top`. The alignment of a text anchor only becomes evident if `mso-fit-text-to-shape` is `false`. This property is different from the `vertical-align` CSS property, which is used for ideographic languages. Allowed values are: • `top` • `middle` • `bottom` • `top-center` • `middle-center` • `bottom-center` • `top-baseline` • `bottom-baseline` • `top-center-baseline` • `bottom-center-baseline`

The following properties are only used by the textpath element (§19.1.2.23):

Property	Description
`font`	Specifies a compound value of font settings. Default is no value. The values are the same as those of the CSS font property. The order of definitions in the string is: font-style, font-variant, font-weight, font-size, line-height, font-family.
`font-family`	Specifies the family of the font. Default is no value. The values are the same as those of the CSS font-family property.
`font-size`	Specifies the size of the font. Default is no value. The font size is defined in points. The values are the same as those of the CSS font-size property.
`font-style`	Specifies the amount of slant for a font. Default is `normal`. The values are the same as those of the CSS font-style property. Allowed values are:

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>

- `normal`
- `italic`
- `oblique` - Treated the same as italic.

</td></tr>
<tr><td>`font-variant`</td><td>Specifies the variant style of a font. Default is `normal`. The values are the same as those of the CSS font-variant property. Allowed values are:

- `normal`
- `small-caps`

</td></tr>
<tr><td>`font-weight`</td><td>Specifies the thickness of the letters of the font. Default is `normal`. The values are the same as those of the CSS font-weight property. Allowed values are:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>`normal`
`lighter`
`100`
`200`
`300`
`400`</td><td>Treated as non-bold.</td></tr>
<tr><td>`bold`
`bolder`
`500`
`600`
`700`
`800`
`900`</td><td>Treated as bold.</td></tr>
</table>

</td></tr>
<tr><td>`mso-text-shadow`</td><td>Specifies whether a shadow is applied to the text on a text path. Default is `false`.</td></tr>
<tr><td>`text-decoration`</td><td>Specifies the style of text decoration. Default is `none`. The values are the same as those of the CSS text-decoration property. Allowed values are:

- `none`

</td></tr>
</table>
</td></tr>
</table>

Attributes	Description
	• underline • overline • line-through • blink
v-rotate-letters	Specifies whether the letters of the text are rotated counterclockwise by 90 degrees. Default is false.
v-same-letter-heights	Specifies whether all letters are the same height regardless of initial case. If true, the lowercase letters are stretched to the height of the uppercase letters. Default is false.
v-text-align	Specifies the alignment of text. Default is left. Allowed values are: • left • right • center • justify • letter-justify - Distributes the extra space between the letters. • stretch-justify - Stretches the letters to fill in the space.
v-text-kern	Specifies whether kerning is turned on. Default is false.
v-text-reverse	Specifies whether the layout order of rows is reversed. Default is false. This is used for vertical text layout.
v-text-spacing-mode	Specifies the mode for letter spacing. Default is tightening. This property determines whether space is removed between each letter (tightening) or added between each letter (tracking). The amount of letter spacing change is defined by the v-text-spacing property. Allowed values are: • tightening • tracking
v-text-spacing	Specifies the amount of spacing for text in 100ths of single line spacing. Default is 100.
	The line (§19.1.2.12), polyline (§19.1.2.15) and curve (§19.1.2.3) elements ignore the following properties: • top • left • width • height

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
The following properties are not inherited by an element that references a shapetype element (§19.1.2.20) via the id attribute:

flip
height
left
margin-left
margin-top
position
rotation
top
visibility
width
z-index

The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>target (Hyperlink Display Target)</td><td>
Specifies a frame or window that a URL is displayed in. Default is no value. Allowed values are:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><targetname></td><td>String containing the name of the frame or window in which to load the document.</td></tr>
<tr><td>_blank</td><td>Specifies that the linked document is loaded into a new blank window. This window is not named.</td></tr>
<tr><td>_media</td><td>Specifies that the linked document is loaded into the browser’s multimedia pane.</td></tr>
<tr><td>_parent</td><td>Specifies that the linked document is loaded into the immediate parent of the document containing the link.</td></tr>
<tr><td>_search</td><td>Specifies that the linked document is loaded into the browser's search pane.</td></tr>
<tr><td>_self</td><td>Specifies that the linked document is loaded into the window in which the link was clicked (the active window).</td></tr>
<tr><td>_top</td><td>Specifies that the linked document is loaded into the topmost window.</td></tr>
</table>

[Example:

<pre>
<v:shape …
  href="http://www.openxmlformats.org"
  target="_self" … >
</pre>
</td></tr>
</table>

Attributes	Description
	`</v:shape>` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

title (Shape Title)	Specifies the text displayed when the mouse pointer moves over the shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … title="tooltip" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

userdrawn (Exists In Master Slide)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the user has added the shape to a master slide. Default is `false`. Used by PresentationML.
	
	[*Example*:
	
	`<v:shape … o:userdrawn="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

userhidden (Hide Script Anchors)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a script anchor is hidden. Default is `false`. If `true`, script anchors stay hidden even if the shape is otherwise visible. A script anchor is the visual representation of a script that when displayed in an application.
	
	[*Example*:
	
	`<v:shape … o:userhidden="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

wrapcoords (Shape Bounding Polygon)	Specifies the bounding polygon that surrounds a shape. This is specified using a comma-delimited list of x and y coordinates: "x1,y1,x2,y2,x3,y3,…" This is used when text is

Attributes	Description

	tightly wrapped around a shape. Default is no value until the mso-wrap-mode style attribute is set to `tight` or `through`.
	
	[*Example*:
	
	`<v:shape …`
	`  wrapcoords="0,0 0,200, 200,200, 200,0" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Arc) is located in §A.6.1. *end note*]

19.1.2.2 background (Document Background)

This element describes the fill of the background of a page using vector graphics fills. Fills consist of simple colors, more advanced effects defined through the fill element (§19.1.2.5), or images.

[*Example*: The following shades the page background a pale red:

```
<v:background fillcolor="#c0504d">
</v:background>
```

This uses the fill element (§19.1.2.5) to create a gradient background fill:

```
<v:background>
 <v:fill type="gradient" color="#c0504d" color2="#f0504d" angle="45"/>
</v:background>
```

end example]

Attributes	Description

bwmode (Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies how a shape renders for black-and-white output devices. When a shape is printed on a black-and-white printer or displayed in a black-and-white view in an application, several options are possible. Default is `auto`, which uses o:bwnormal for normal black-and-white rendering and o:bwpure for pure black-and-white rendering
	
	[*Example*: This shape renders in grayscale in a black-and-white environment:
	
	`<v:shape … o:bwmode="grayscale" … >`
	`</v:shape>`

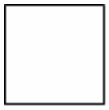
Attributes	Description
	end example] The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

bwnormal (Normal Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the black-and-white mode for normal black-and-white output devices. Default is `auto`.
	
	[*Example*: This shape renders in a pale grayscale in a normal black-and-white environment:
	
	`<v:shape …`
	`  o:bwmode="lightgrayscale" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

bwpure (Pure Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the black-and-white mode for pure black-and-white output devices. Default is `auto`.
	
	[*Example*: This shape renders in high contrast when in a pure black-and-white environment:
	
	`<v:shape … o:bwmode="highcontrast" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

fillcolor (Fill Color)	Specifies the color to use for the fill. Default is `white`. If the fill element (§19.1.2.5) is present, its color attribute takes precedence. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*: This shape is red if its fill is visible:
	
	`<v:shape … fillcolor="red" … >`
	`</v:shape>`
	
	This is equivalent to:
	
	`<v:shape … fillcolor="#ff0000" … >`

Attributes	Description
	`</v:shape>` *end example*] The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

filled (Shape Fill Toggle)	Specifies whether the closed path is filled. Default is `true`. This attribute is overridden by the fill on attribute.
	
	[*Example*:
	
	`<v:shape … filled="f"`
	`  fillcolor="red" …>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

id (Unique Identifier)	Specifies a unique identifier that can be used to reference a VML object.
	
	Default is no value.
	
	[*Example*:
	
	`<v:shape … id="myShape" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

targetscreensize (Target Screen Size) Namespace: urn:schemas-microsoft-com:office:office	Specifies the target resolution used for WordprocessingML documents with a gradient or picture filled background. Default is no value. Allowed values are: • `544,376` • `640,480` • `720,512` • `800,600` • `1024,768` • `1152,862`

Attributes	Description
	The possible values for this attribute are defined by the ST_ScreenSize simple type (§19.2.3.23).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Background) is located in §A.6.1. *end note*]

19.1.2.3 curve (Bezier Curve)

This element is used to draw a cubic bézier curve.

The following properties of the style attribute are ignored:

- `top`
- `margin-top`
- `center-y`
- `left`
- `margin-left`
- `center-x`
- `width`
- `height`

[*Example*: The following specifies a simple curve that opens upward:

```
<v:curve id="mycurve"
from="10pt,10pt" to="100pt,10pt"
control1="40pt,30pt" control2="85pt,30pt">
</v:curve>
```

This shape is created:

end example]

Attributes	Description

allowincell (Allow in Table Cell)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can be placed in a table. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:allowincell="true" … >`
	`</v:shape>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

allowoverlap (Allow Shape Overlap)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can overlap another shape. Default is `true`. If `false`, the shape shifts left or right so as not to overlap another shape, similar to the behavior of the HTML float attribute.
	
	[*Example*:
	
	`<v:shape … o:allowoverlap="false" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

alt (Alternate Text)	Specifies alternative text describing the graphical object. This text should provide a brief description of the shape for use by accessibility tools. Default is no value.
	
	[*Example*: The alt text describes the basic shape:
	
	`<v:shape … fillcolor="red"`
	`  alt="Red rectangle">`
	`</v:shape>`
	
	The alt text describes the contents of a shape displaying an image:
	
	`<v:shape … alt="Picture of a sunset">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderbottomcolor (Bottom Border Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the bottom border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderbottomcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderleftcolor	Specifies the left border color of an inline shape. Default is no value.

Attributes	Description

(Border Left Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	[*Example*:
	
	`<v:shape … o:borderleftcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderrightcolor (Border Right Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the right border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderrightcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

bordertopcolor (Border Top Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the top border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:bordertopcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

bullet (Graphical Bullet)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the shape is a graphical bullet. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:bullet="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

button (Button Behavior Toggle)	Specifies whether a shape exhibits button press behavior on click. Default is `false`.

Attributes	Description

Namespace: urn:schemas-microsoft-com:office:office	[*Example*:
	
	`<v:shape … o:button="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

bwmode (Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies how a shape renders for black-and-white output devices. When a shape is printed on a black-and-white printer or displayed in a black-and-white view in an application, several options are possible. Default is `auto`, which uses o:bwnormal for normal black-and-white rendering and o:bwpure for pure black-and-white rendering.
	
	bwnormal and bwpure are subordinate to bwmode. If bwmode is "auto" then the value for bwnormal or bwpure is used depending on what the output format is. An application can define for itself what, if any, difference there is between normal B&W and pure B&W. [*Example*: Normal B&W might allow greyscale and pure B&W might not. *end example*]
	[*Example*: This shape renders in grayscale in a black-and-white environment:
	
	`<v:shape … o:bwmode="grayscale" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

bwnormal (Normal Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the black-and-white mode for normal black-and-white output devices. Default is `auto`.
	
	[*Example*: This shape renders in a pale grayscale in a normal black-and-white environment:
	
	`<v:shape …  o:bwmode="auto" o:bwnormal="lightgrayscale" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

bwpure (Pure Black-and-White Mode)	Specifies the black-and-white mode for pure black-and-white output devices. Default is `auto`. [*Example*: This shape renders in high contrast when in a pure black-and-white

Attributes	Description

Namespace: urn:schemas-microsoft-com:office:office	environment:
	
	`<v:shape … o:bwmode="auto" o:bwpure="highcontrast" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

chromakey (Image Transparency Color)	Specifies a color value that is transparent and show anything behind the shape. Default is no value.
	
	[*Example*:
	
	`<v:image … chromakey="white" …>`
	`</v:image>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

class (CSS Reference)	Specifies a reference to the definition of a CSS style. Default is no value.
	
	[*Example*: The snippets below are equivalent:
	
	`…`
	`.narrowstyle {width:50;height:100}`
	`…`
	`<v:shape … class="narrowstyle"`
	`  style="top:1;left:1">`
	`</v:shape>`
	
	`<v:shape … style="top:1;left:1;`
	`  width:50;height:100">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

clip (Clipping Toggle) Namespace: urn:schemas-	Specifies that the clipping region is active. This is used in conjunction with the clippath (§19.2.2.3) element to create a clipping region. [*Example*:

Attributes	Description

microsoft-com:office:office	`<v:shape … o:clip="true">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

cliptowrap (Clip to Wrapping Polygon)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the clipping region for the shape aligns with the wrapping polygon that tightly surrounds the entire shape (essentially, that the shape shall not be drawn beyond its wrapping polygon's extents – if it does, the shape shall be clipped). Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:cliptowrap="true">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

connectortype (Shape Connector Type)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of connector used for joining shapes. Default is `straight`.
	
	[*Example*:
	
	`<v:shape … o:connectortype="elbow" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ConnectorType simple type (§19.2.3.7).

control1 (First Curve Control Point)	Specifies the first control point for the curve, given in the coordinate space of the parent element. Default is "10,10". If the parent is not a VML element, the default unit is a pixel. Allowed units are in, cm, mm, pt, pc and px.
	
	[*Example*:
	
	`<v:curve … control1="20,30" … >`
	`</v:curve>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

Attributes	Description

control2 (Second Curve Control Point)	Specifies the second control point for the curve, given in the coordinate space of the parent element. Default is "20,0". If the parent is not a VML element, the default unit is a pixel. Allowed units are in, cm, mm, pt, pc and px.
	
	[*Example*:
	
	`<v:curve … control2="50,20" … >`
	`</v:curve>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

coordorigin (Coordinate Space Origin)	Specifies the coordinate of the top left corner of the shape's coordinate space. This determines the position of the (0,0) coordinate space origin within the shape's bounding box. Default is "0,0", which places the (0,0) origin at the top left corner of the bounding box.
	
	This affects shape properties that specify coordinate positions, such as the path attribute. Thus a path can be defined against a generic (0,0) origin and the coordorigin value translates the entire path within the shape's bounding space.
	
	[*Example*: The horizontal and vertical coordinate space ranges from -100 to +100 because the coordinate space (coordsize) is 200 by 200 and the top left coordinate is (-100,-100). The (0,0) origin lies at the center of the shape's bounding box, as evidenced by the position of the shape's path within the coordinate space:
	
	`<v:shape … coordsize="200,200"`
	`coordorigin="-100,-100"`
	`path="m 0,0 l 0,50, 50,50, 50,0 x e">`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

coordsize (Coordinate Space Size)	Specifies the size of the shape's coordinate space in coordinate units. Default is "1000,1000". The physical size of a coordinate unit length is determined by both the size of the coordinate space (coordsize) and the size of the shape (style width and height). The coordsize attribute defines the number of horizontal and vertical subdivisions into which

Attributes	Description

	the shape's bounding box is divided. The combination of coordsize and style width/height effective scales the shape anisotropically.
	
	[*Example*: The path is 50 units wide and tall, which is 25% of the size of the coordinate space:
	
	`<v:shape … coordsize="200,200"`
	`  coordorigin="-100,-100"`
	`  path="m 0,0 l 0,50, 50,50, 50,0 x e">`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

dgmlayout (Diagram Node Layout Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of automatic layout to apply to the diagram node. This is only meaningful if the shape is a node in an organization chart, which is denoted by the `orgchart` value of the editas attribute of the group element.
	[*Example*:
	
	`<v:shape … dgmlayout="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).

dgmlayoutmru (Diagram Node Recent Layout Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of automatic layout most recently used on the child elements of the diagram node. This is only meaningful if the shape is a node in anorganization chart, which is denoted by the `orgchart` value of the editas attribute of the group element.
	
	[*Example*:
	
	`<v:shape … dgmlayout="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).

Attributes	Description

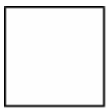
dgmnodekind (Diagram Node Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies an optional, application-defined parameter that is intended to be used by the application to tag different types of nodes in a diagram.
	
	[*Example*:
	
	`<v:shape … dgmnodekind="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

doubleclicknotify (Double-click Notification Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that an event message is sent when a shape is double-clicked. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:doubleclicknotify="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

fillcolor (Fill Color)	Specifies the color to use for the fill. Default is `white`. If the fill element (§19.1.2.5) is present, its color attribute takes precedence. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*: This shape is red if its fill is visible:
	
	`<v:shape … fillcolor="red" … >`
	`</v:shape>`
	
	This is equivalent to:
	
	`<v:shape … fillcolor="#ff0000" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

filled (Shape Fill Toggle)	Specifies whether the closed path is filled. Default is `true`. This attribute is overridden by the fill on attribute.

Attributes	Description

	[*Example*:
	
	`<v:shape … filled="f"`
	`  fillcolor="red" …>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

forcedash (Force Dashed Outline)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a dashed outline is used to draw a shape when a shape has no line or fill. Default is `false`.
	
	Used by PresentationML placeholders to draw a dashed outline when there is no line and no fill for a shape.
	
	[*Example*:
	
	`<v:shape … o:forcedash="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

from (Curve Starting Point)	Specifies the starting point of the line in the coordinate space of the parent element. Default is "0,0". If the parent is not a VML element, the default unit is a pixel. Allowed units are in, cm, mm, pt, pc and px.
	
	[*Example*:
	
	`<v:curve … from="10,10" … >`
	`</v:curve>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

hr (Horizontal Rule Toggle)	Specifies that a shape is a horizontal rule. Default is `false`.

Attributes	Description

Namespace: urn:schemas-microsoft-com:office:office	[*Example*:
	
	`<v:shape … o:hr="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hralign (Horizontal Rule Alignment)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the alignment of a horizontal rule. Default is `left`.
	
	[*Example*:
	
	`<v:shape … o:hralign="center" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_HrAlign simple type (§19.2.3.16).

href (Hyperlink Target)	Specifies a hyperlink URL target for the shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … href="http://www.openxmlformats.org" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

hrnoshade (Horizontal Rule 3D Shading Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the horizontal rule does not have 3-D shading. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hrnoshade="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hrpct (Horizontal Rule Length Percentage)	Specifies the length of a horizontal rule as a percentage of page width. Default is 0. [*Example*:

Attributes	Description

Namespace: urn:schemas-microsoft-com:office:office	`<v:shape … o:hrpct="85" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema float datatype.

hrstd (Horizontal Rule Standard Display Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape is displayed as a standard horizontal rule. Only applies if hr is `true`. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hrstd="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

id (Unique Identifier)	Specifies a unique identifier that can be used to reference a VML object.
	
	Default is no value.
	
	[*Example*:
	
	`<v:shape … id="myShape" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

insetmode (Text Inset Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the application calculates the internal text margin instead of using the inset attribute. Default is `custom`. This attribute is only meaningful for text boxes.
	
	[*Example*:
	
	`<v:shape … o:insetmode="auto" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_InsetMode simple type (§19.2.3.17).

insetpen (Inset	Specifies that the border shall be displayed inside of the path defining the shape, rather

Attributes	Description

Border From Path)	than along the path (the default border placement), or outside of the path as might be done with an image.
	
	[*Example*:
	
	`<v:shape … insetpen="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

ole (Embedded Object Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the shape is an embedded object. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:ole="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

oleicon (Embedded Object Icon Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether an embedded object is displayed as an icon. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oleicon="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

oned (Shape Handle Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the extra handles of a shape are hidden. If `true`, hides all shape handles except the top left and bottom right; that is, the same handles that are used for a straight line segment. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oned="true" … >`
	`</v:shape>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

opacity (Fill Color Opacity)	Specifies the opacity of the primary fill color. Default is 1.0. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. [*Example*: A value of "52429f" represents 52429/65536 or 0.8. *end example*]
	
	[*Example*: The red color is 25% opaque:
	
	`<v:fill type="gradient" color="red"`
	`  color2="blue" opacity=".25">`
	`</v:fill>`
	
	  `opacity="1"`
	  `opacity=".25"`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

preferrelative (Relative Resize Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the original size of an object is saved after reformatting. Default is `false`. If `true`, the original size of the object is stored and all resizing is based on a percentage of that original size. Otherwise, each resizing resets the scale to 100%.
	
	[*Example*:
	
	`<v:shape … o:preferrelative="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

print (Print Toggle)	Specifies whether the shape is printed. Default is `true`.
	
	[*Example*:
	
	`<v:shape … print="false" … >`
	`</v:shape>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

regroupid (Regroup ID)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies a previous group for a shape. An ID number is used to identify groups of shapes that are no longer grouped. This allows shapes to be regrouped programmatically.
	
	[*Example*: The shape was part of a group identified by the ID 040754:
	
	`<v:shape … o:regroupid="040754" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

spid (Optional String) Namespace: urn:schemas-microsoft-com:office:office	Specifies an optional string that an application can use to Identify the particular shape. Default is no value. The possible values for this attribute are defined by the W3C XML Schema string datatype.
spt (Optional Number) Namespace: urn:schemas-microsoft-com:office:office	Specifies an optional number that an application can use to associate the particular shape with a defined shape type. Default is 0. The possible values for this attribute are defined by the W3C XML Schema float datatype.

strokecolor (Shape Stroke Color)	Specifies the primary color of the brush to use to stroke the path of the shape. Default is `black`. The color attribute of the stroke element (§19.1.2.21) overrides this. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*:
	
	`<v:shape … strokecolor="red" …>`
	`</v:shape>`
	
	 
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

stroked (Shape Stroke Toggle)	Specifies whether the path defining the shape is stroked with a solid line. The stroke element (§19.1.2.21) defines other strokes. The on attribute of the stroke element overrides this attribute. Default is true.
	
	[*Example*:
	
	`<v:shape … fillcolor="red"`
	`  stroked="false" strokecolor="blue"…>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

strokeweight (Shape Stroke Weight)	Specifies the width of the brush to use to stroke the path. Default is 1 point. If a number is given without units, the emu is used. The weight attribute of the stroke element (§19.1.2.21) overrides this attribute.
	
	[*Example*:
	
	`<v:shape … strokeweight="3pt" … >`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

style (Shape Styling Properties)	Specifies the CSS2 styling properties of the shape. The CSS2 (Cascading Style Sheets, Level 2) specification, a Recommendation of the World Wide Web Consortium, is available here: http://www.w3.org/TR/REC-CSS2. This attribute uses a semi-colon delimited list of “name:value” pairs, the syntax defined by section 4.1.8 of the CSS2 specification for grouped declarations (without the surrounding braces). The following tables define the allowed properties and the VML treatment of each.

Attributes	Description

	[*Example*:
	`<v:shape … style='position:absolute;width:100pt;height:50pt' …`
	`>`
	`</v:shape>`
	end example]

Property	Description
`flip`	Specifies that the orientation of a shape is flipped. Default is no value. Allowed values are: • x - Flip along the y-axis, reversing the x-coordinates. • y - Flip along the x-axis, reversing the y-coordinates. • xy - Flip along both the y- and x-axis. • yx - Flip along both the x- and y-axis.
`height`	Specifies the height of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
`left`	Specifies the position of the left of the containing block of the shape relative to the element left of it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
`margin-bottom`	Specifies the position of the bottom of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are:

Attributes	Description	
		• auto - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	margin-left	Specifies the position of the left of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • auto - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	margin-right	Specifies the position of the right of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • auto - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	margin-top	Specifies the position of the top of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • auto - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	mso-position-	Specifies the horizontal positioning data for objects in

Attributes	Description
	`horizontal` — WordprocessingML documents. Default is `absolute`. Allowed values are: • `absolute` • `left` • `center` • `right` • `inside` • `outside`
	`mso-position-horizontal-relative` — Specifies relative horizontal position data for objects in WordprocessingML documents. This modifies the `mso-position-horizontal` property. Default is `text`. Allowed values are: • `margin` • `page` • `text` • `char`
	`mso-position-vertical` — Specifies the vertical positioning data for objects in WordprocessingML documents. Default is `absolute`. Allowed values are: • `absolute` • `top` • `center` • `bottom` • `inside` • `outside`
	`mso-position-vertical-relative` — Specifies relative vertical position data for objects in WordprocessingML documents. This modifies the `mso-position-vertical` property. Default is `text`. Allowed values are: • `margin` • `page` • `text` • `line`
	`mso-wrap-distance-bottom` — Specifies the distance from the bottom of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change

Attributes	Description
	the origin.
`mso-wrap-distance-left`	Specifies the distance from the left side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
`mso-wrap-distance-right`	Specifies the distance from the right side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
`mso-wrap-distance-top`	Specifies the distance from the top of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
`mso-wrap-edited`	Specifies whether the wrap coordinates were customized by the user. If the wrap coordinates are generated by an editor, this property is `true`; otherwise they were customized by a user. Default is `false`.
`mso-wrap-style`	Specifies the wrapping mode for text in shapes in WordprocessingML documents. Default is `square`. Allowed values are: • `square` - Wraps text inside the shape in a square. • none - Text does not wrap.
`position`	Specifies the kind of positioning used to place an element. Default is `static`. When the element is contained inside a group, this property shall be `absolute`. Allowed values are: • `static` - The element is positioned according to the normal flow of the page. The `top` and `left` properties are ignored. If the object is anchored inline, this value is used. • `absolute` - The element is positioned relative to the parent, using the `top` and `left` properties. • `relative` - The element is positioned according to the normal flow of the page, but the `top` and `left` properties are used. The overlap of overlapping elements is governed by the `z-index` property.
`rotation`	Specifies the angle that a shape is rotated, in degrees. Default is 0. Positive angles are clockwise.

Attributes	Description	
	`top`	Specifies the position of the top of the containing block of the shape relative to the element above it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`visibility`	Specifies whether a shape is displayed. Only `inherit` and `hidden` are used; any other values are mapped to `inherit`. Default is `inherit`. Allowed values are: • `hidden` - The shape is not visible, but is still part of the flow of the objects in the browser. Mouse events are not processed. • `inherit` - The visibility state is inherited from the parent of the shape.
	`width`	Specifies the width of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`z-index`	Specifies the display order of overlapping shapes. Default is 0. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Uses the order that the shapes appear in the page, bottom to top. • <order>- A number that represents the stacking precedence. Shapes with higher numbers are placed on top of those with lower numbers. Negative numbers are allowed.

Attributes	Description
	The following properties are only used by the textbox element (§19.1.2.22):

Property	Description
`direction`	Specifies the direction of the text in the textbox. Default is `ltr`. This property is superceded by the `mso-direction-alt` property if that is specified. Allowed values are: • `ltr` - Text is displayed left-to-right. • `rtl` - Text is displayed right-to-left.
`layout-flow`	Determines the flow of the text layout in a textbox. Default is `horizontal`. Allowed values are: • `horizontal` - Text is displayed horizontally. • `vertical` - Text is displayed vertically. • `vertical-ideographic` - Ideographic text is displayed vertically. • `horizontal-ideographic` - Ideographic text is displayed horizontally.
`mso-direction-alt`	Specifies an alternate direction for text in textboxes. Overrides the `direction` property. The only allowed value is `context`.
`mso-fit-shape-to-text`	Specifies whether the shape stretches to fit the text in the textbox. Default is `false`.
`mso-fit-text-to-shape`	Specifies whether the text stretches to fit the textbox. Default is `false`.
`mso-layout-flow-alt`	Specifies the alternate layout flow for text in textboxes. This property is used instead of `layout-flow` when the layout flow is from bottom to top for non-ideographic languages. Its only value is `bottom-to-top`.
`mso-next-textbox`	Specifies the ID of the next textbox in a series. Used to keep track of a set of linked textboxes. Default is no value.
`mso-rotate`	Specifies a specific rotation value for text in a textbox. Default is 0. Allowed values are: • 0 • 90 • 180

Attributes	Description

	• -90
mso-text-scale	Specifies the scaling factor for fitting text to shapes. Default is 0. This property is only used if mso-fit-text-to-shape is true.
v-text-anchor	Specifies the vertical anchoring of text in a textbox. Default is top. The alignment of a text anchor only becomes evident if mso-fit-text-to-shape is false. This property is different from the vertical-align CSS property, which is used for ideographic languages. Allowed values are: • top • middle • bottom • top-center • middle-center • bottom-center • top-baseline • bottom-baseline • top-center-baseline • bottom-center-baseline

The following properties are only used by the textpath element (§19.1.2.23):

Property	Description
font	Specifies a compound value of font settings. Default is no value. The values are the same as those of the CSS font property. The order of definitions in the string is: font-style, font-variant, font-weight, font-size, line-height, font-family.
font-family	Specifies the family of the font. Default is no value. The values are the same as those of the CSS font-family property.
font-size	Specifies the size of the font. Default is no value. The font size is defined in points. The values are the same as those of the CSS font-size property.
font-style	Specifies the amount of slant for a font. Default is normal. The values are the same as those of the CSS font-style property. Allowed values are: • normal • italic • oblique - Treated the same as italic.
font-variant	Specifies the variant style of a font. Default is normal. The values are the same as those of the CSS font-variant property. Allowed values are:

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>

- normal
- small-caps

</td></tr>
<tr><td>font-weight</td><td>Specifies the thickness of the letters of the font. Default is normal. The values are the same as those of the CSS font-weight property. Allowed values are:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>normal
lighter
100
200
300
400</td><td>Treated as non-bold.</td></tr>
<tr><td>bold
bolder
500
600
700
800
900</td><td>Treated as bold.</td></tr>
</table>
</td></tr>
<tr><td>mso-text-shadow</td><td>Specifies whether a shadow is applied to the text on a text path. Default is false.</td></tr>
<tr><td>text-decoration</td><td>Specifies the style of text decoration. Default is none. The values are the same as those of the CSS text-decoration property. Allowed values are:

- none
- underline
- overline
- line-through
- blink

</td></tr>
<tr><td>v-rotate-</td><td>Specifies whether the letters of the text are rotated counterclockwise by 90 degrees. Default is false.</td></tr>
</table>
</td></tr>
</table>

Attributes	Description
	letters	
	v-same-letter-heights	Specifies whether all letters are the same height regardless of initial case. If true, the lowercase letters are stretched to the height of the uppercase letters. Default is false.
	v-text-align	Specifies the alignment of text. Default is left. Allowed values are: • left • right • center • justify • letter-justify - Distributes the extra space between the letters. • stretch-justify - Stretches the letters to fill in the space.
	v-text-kern	Specifies whether kerning is turned on. Default is false.
	v-text-reverse	Specifies whether the layout order of rows is reversed. Default is false. This is used for vertical text layout.
	v-text-spacing-mode	Specifies the mode for letter spacing. Default is tightening. This property determines whether space is removed between each letter (tightening) or added between each letter (tracking). The amount of letter spacing change is defined by the v-text-spacing property. Allowed values are: • tightening • tracking
	v-text-spacing	Specifies the amount of spacing for text in 100ths of single line spacing. Default is 100.

The line (§19.1.2.12), polyline (§19.1.2.15) and curve (§19.1.2.3) elements ignore the following properties:

- top
- left
- width
- height

The following properties are not inherited by an element that references a shapetype element (§19.1.2.20) via the id attribute:

- flip

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>

height
left
margin-left
margin-top
position
rotation
top
visibility
width
z-index

The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>target (Hyperlink Display Target)</td><td>
Specifies a frame or window that a URL is displayed in. Default is no value. Allowed values are:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><targetname></td><td>String containing the name of the frame or window in which to load the document.</td></tr>
<tr><td>_blank</td><td>Specifies that the linked document is loaded into a new blank window. This window is not named.</td></tr>
<tr><td>_media</td><td>Specifies that the linked document is loaded into the browser’s multimedia pane.</td></tr>
<tr><td>_parent</td><td>Specifies that the linked document is loaded into the immediate parent of the document containing the link.</td></tr>
<tr><td>_search</td><td>Specifies that the linked document is loaded into the browser's search pane.</td></tr>
<tr><td>_self</td><td>Specifies that the linked document is loaded into the window in which the link was clicked (the active window).</td></tr>
<tr><td>_top</td><td>Specifies that the linked document is loaded into the topmost window.</td></tr>
</table>
[Example:
<pre>
<v:shape …
  href="http://www.openxmlformats.org"
  target="_self" … >
</v:shape>
</pre>
end example]
The possible values for this attribute are defined by the W3C XML Schema string
</td></tr>
</table>

Attributes	Description
	datatype.

title (Shape Title)	Specifies the text displayed when the mouse pointer moves over the shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … title="tooltip" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

to (Curve Ending Point)	Specifies the ending point of the line in the coordinate space of the parent element. Default is "30,20". If the parent is not a VML element, the default unit is a pixel. Allowed units are in, cm, mm, pt, pc and px.
	
	[*Example*:
	
	`<v:curve … to="40,40" … >`
	`</v:curve>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

userdrawn (Exists In Master Slide)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the user has added the shape to a master slide. Default is `false`. Used by PresentationML.
	
	[*Example*:
	
	`<v:shape … o:userdrawn="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

userhidden (Hide Script Anchors) Namespace: urn:schemas-microsoft-com:office:office	Specifies whether a script anchor is hidden. Default is `false`. If `true`, script anchors stay hidden even if the shape is otherwise visible. A script anchor is the visual representation of a script that when displayed in an application. [*Example*: `<v:shape … o:userhidden="true" … >`

Attributes	Description
	`</v:shape>` *end example*] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

wrapcoords (Shape Bounding Polygon)	Specifies the bounding polygon that surrounds a shape. This is specified using a comma-delimited list of x and y coordinates: "x1,y1,x2,y2,x3,y3,…" This is used when text is tightly wrapped around a shape. Default is no value until the mso-wrap-mode style attribute is set to `tight` or `through`.
	
	[*Example*:
	
	`<v:shape …`
	`wrapcoords="0,0 0,200, 200,200, 200,0" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Curve) is located in §A.6.1. *end note*]

19.1.2.4 f (Single Formula)

This element defines a single value as the result of the evaluation of an expression. The expression is defined by the eqn attribute and has the general form of an operation followed by up to three arguments, which consist of adjustment values (see the adj attribute of the shape element (§19.1.2.19)), the results of earlier formulas, fixed numbers or pre-defined values. Each f value is referenced using "@" followed by a number corresponding to the zero-based index for that value in the list of f elements. [*Example*: For example, the value of the second f element is referenced as "@2". *end example*]

[*Example*: The following defines a blue arrow pointing to the right:

```
<v:shape coordsize="21600,21600" adj="18000,5400,10800"
  path="m @0,0 l @0,@1 0,@1 0,@3 @0,@3 @0,21600 21600,10800 x e"
  style='left:50pt;top:50pt;width:90pt;height:30pt'
  fillcolor="#4f81bd" strokecolor="#4f81bd" strokeweight="2pt">
  <v:formulas>
   <v:f eqn="val #0"/>
   <v:f eqn="val #1"/>
   <v:f eqn="val #2"/>
```

```
    <v:f eqn="sum height 0 #1"/>
    <v:f eqn="sum #2 0 #1"/>
    <v:f eqn="sum width 0 #0"/>
    <v:f eqn="prod @5 @4 #2"/>
    <v:f eqn="sum width 0 @6"/>
   </v:formulas>
  </v:shape>
```

The shape looks like this:

end example]

Attributes	Description
eqn (Equation)	Specifies a single formula, which consists of a named operation followed by up to three parameters, typically described as v, P1 and P2. Up to 128 formulas can be specified. These operations are defined (calculation accuracy is discussed below):

Operation	Description
`val`	v Returns the supplied value. Exact.
`sum`	$\mathrm{v} + \mathrm{P1} - \mathrm{P2}$ Addition and subtraction. Exact.
`product`	$\mathrm{v} \times \mathrm{P1}/\mathrm{P2}$ Multiplication and division. Rounds up.
`mid`	$(\mathrm{v} + \mathrm{P1})/2$ Simple average. Rounds toward zero.
`abs`	$\lvert\mathrm{v}\rvert$ Absolute value. Exact.
`min`	$\min(\mathrm{v}, \mathrm{P1})$ The lesser of two values. Exact.
`max`	$\max(\mathrm{v}, \mathrm{P1})$ The greater of two values. Exact.
`if`	$\mathrm{v} > 0\ ?\ P1 : P2$ Conditional selection. Exact.
`mod`	$\sqrt{\mathrm{v}^2 + \mathrm{P1}^2 + \mathrm{P1}^2}$ Modulus. Inexact.
`atan2`	$\mathrm{atan2}(\mathrm{P1}, \mathrm{v})$ Trigonometric arc tangent of a quotient. Result is in "fd" units or fractional degrees - degrees $\times\ 2^{16}$.

Attributes	Description

	Inexact.
`sin`	$\mathrm{v} \times \sin(\mathrm{P1})$ Sine. Argument is in "fd" units or fractional degrees - degrees $\times 2^{16}$. Inexact.
`cos`	$\mathrm{v} \times \cos(\mathrm{P1})$ Cosine. Argument is in "fd" units or fractional degrees - degrees $\times 2^{16}$. Inexact.
`cosatan2`	$\mathrm{v} \times \cos\big(\mathrm{atan2}(\mathrm{P2}, \mathrm{P1})\big)$ Preserves full accuracy in the intermediate calculation. Inexact.
`sinatan2`	$\mathrm{v} \times \sin\big(\mathrm{atan2}(\mathrm{P2}, \mathrm{P1})\big)$ Preserves full accuracy in the intermediate calculation. Inexact.
`sqrt`	$\sqrt{\mathrm{v}}$ Square root. Result is positive and rounds down. Inexact.
`sumangle`	$\mathrm{v} + \mathrm{P1} \times 2^{16} - \mathrm{P2} \times 2^{16}$ Adds an existing angle in fd units (v) to two other angles specified in degrees. P1 and P2 are scaled by 2^{16}. Exact.
`ellipse`	$\mathrm{P2}\sqrt{1 - \left(\frac{\mathrm{v}}{\mathrm{P1}}\right)^2}$ The eccentricity formula for an ellipse, where v is length of the semiminor axis and P1 is the length of the semimajor axis. Inexact.
`tan`	$\mathrm{v} \times \tan(\mathrm{P1})$ Tangent. Argument is in "fd" units or fractional degrees - degrees $\times 2^{16}$. Inexact.

Formulas are evaluated to full precision, but the result is always a 32-bit integer. Formula authors should avoid formulas which are discontinuous - not only are many of the trigonometric operations inexact, the transformations within the coordinate spaces are also inexact. This can mean that a set of formulas which is discontinuous evaluates to give very different path values with the same input on two different systems.

When an operation is marked as exact then a conforming implementation shall always generate the correct arithmetic answer (unless the calculations overflow internally). The product operation is required to round to the nearest integer. If the result is exactly 0.5 then it shall be rounded up to the next numerically greater integer. The mid operation is required to round towards 0.

All other operations are inexact, but the implementation shall round non-integral values down (towards -infinity) and should perform internal calculations with this form of

Attributes	Description
	rounding. The arguments used in the evaluation of a formula are normally either fixed numbers, the result of the evaluation of a previous formula or an adjust value - the value of the corresponding entry in the shape adj attribute. Fixed numbers shall be positive integral values in the range 0 to 65535 (unsigned 16-bit numbers). The following named values are defined:

Value	Description
`@n`	The value of formula n, where n is the zero-based index of the formula in the list of formulas. n shall be less than the current formula index.
`#n`	Adjustment (adj) value n. n shall be in the range 0 to 7.
`width`	The width defined by the coordsize attribute.
`height`	The height defined by the coordsize attribute.
`xcenter`	The x ordinate of the center of the coordinate space defined by coordorigin and coordsize.
`ycenter`	The y ordinate of the center of the coordinate space defined by coordorigin and coordsize.
`xlimo`	The x value of the limo attribute (see also the path element (§19.1.2.14)).
`ylimo`	The y value of the limo attribute (see also the path element (§19.1.2.14)).
`hasstroke`	1 if the shape has a stroke operation, 0 if it does not, as determined by the on attribute of the stroke element (§19.1.2.21).
`hasfill`	1 if the shape has a fill operation, 0 if it does not, as determined by the on attribute of the fill element (§19.1.2.5).
`pixellinewidth`	The line width in output device pixels. This is used to outset lines from the edge of a rectangle on the assumption that the implementation draws to lower right pixel in preference to the upper left pixel when a line is on a pixel boundary.
`pixelwidth`	The width of the shape in device pixels (i.e., the coordsize width transformed into device space).

Attributes	Description
	<table><tr><td>pixelheight</td><td>The height of the coordsize in device pixels.</td></tr><tr><td>emuwidth</td><td>The width of the coordsize in EMUs.</td></tr><tr><td>emuheight</td><td>The height of the coordsize in EMUs.</td></tr><tr><td>emuwidth2</td><td>Half the width of the coordsize in EMUs.</td></tr><tr><td>emuheight2</td><td>Half the height of the coordsize in EMUs.</td></tr></table> The EMU, or English Metric Unit, is the smallest unit of measure in VML and corresponds to 914400 EMU per inch or 12700 EMU per point. See above for an example. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_F) is located in §A.6.1. *end note*]

19.1.2.5 fill (Shape Fill Properties)

This element specifies how the path should be filled if something beyond a solid color fill is desired. The attributes of the fill element can be used to describe a powerful set of image- or gradient-based fill patterns. Extensions to the VML fill definition are encoded as sub-elements of fill.

Attributes	Description

alignshape (Align Image With Shape)	Specifies whether an image aligns with the shape. Default is `true`.
	
	[*Example*: The image displayed in the shape is not rotated even though the shape is rotated 30 degrees:
	
	`<v:shape coordorigin="0,0"`
	`coordsize="200,200"`
	`style="top:1;left:1;width:50;`
	`height:50;rotation:30"`
	`path="m 1,1 l 1,200, 200,200, 200,1 x e">`
	`<v:fill alignshape="false" type="frame"`
	`src="myimage.gif">`
	`</v:fill>`
	`</v:shape>`
	
	Applied to a simple square the fill looks like this:

Attributes	Description
	 *end example*] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

althref (Alternate Image Reference Location)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Defines an alternate reference for an image in Macintosh PICT format.
	
	[*Example*:
	
	`<v:fill … althref="myimage.pcz" … >`
	`</v:fill>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

angle (Gradient Angle)	Specifies the direction of a gradient. The vector of a gradient is perpendicular to the vector of the blend direction from one color to another. The default value is zero degrees, which is a horizontal vector from left to right. Positive angles rotate the gradient in a counter-clockwise direction.
	
	[*Example*: The fill is composed of a 45-degree gradient of two colors. Blue is in the top left corner and red is in the bottom right corner.
	
	`<v:fill type="gradient" color="red"`
	`  color2="blue" angle="45">`
	`</v:fill>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema decimal datatype.

aspect (Image Aspect Ratio)	Specifies how the fill image aspect ratio is preserved. Default is `ignore`. Allowed values are: • `ignore` - Ignore aspect ratio. • `atleast` - At least as large as defined by the size attribute.

Attributes	Description

	• `atmost` - No larger than that defined by the size attribute.
	
	In each case, the size attribute is adjusted to preserve the aspect ratio of the image.
	
	[*Example*: The image that makes up the fill is no larger than 20 points by 20 points, limiting the size of the tiles inside the shape.
	
	`<v:fill type="tile" aspect="atmost"`
	`  size="20pt,20pt" src="myimage.gif">`
	`</v:fill>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_ImageAspect simple type (§19.1.3.5).

color (Primary Color)	Specifies the main fill color; functions the same as the fillcolor attribute of the shape element (§19.1.2.19). This attribute overrides the shape's fillcolor. Default is `white`. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*: The shape is blue:
	
	`<v:shape … fillcolor="red" … >`
	`  <v:fill color="blue"/>`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

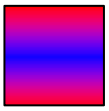
color2 (Secondary Color)	Specifies the secondary fill color, used when a fill type is a pattern or a gradient. Default is `white`. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*: The shape is filled with a horizontal gradient with red at the bottom and blue on top:
	
	`<v:fill type="gradient"`
	`  color="red" color2="blue">`
	`</v:fill>`

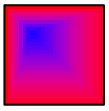
Attributes	Description
	end example] The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

colors (Intermediate Colors)	Specifies an array of comma-separated percentage-color pairs that define intermediate colors and their positions in a gradient fill. The primary color, specified either by the fillcolor attribute of the shape element (§19.1.2.19) or the color attribute of the fill element (§19.1.2.5), is used at the 0% endpoint. The secondary color, specified by the color2 attribute of the fill element (§19.1.2.5), is used at the 100% endpoint. The numeric values can also be specified in 1/65536-ths if a trailing "f" is supplied. [*Example*: For example, a value of "52429f" represents 52429/65536 or 0.8. *end example*]
	
	[*Example*: The shape is filled with a horizontal gradient colored, from bottom to top, red, yellow, green, blue:
	
	`<v:fill type="gradient"`
	`  color="red" color2="blue"`
	`  colors="30% yellow,70% green">`
	`</v:fill>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

detectmouseclick (Detect Mouse Click) Namespace: urn:schemas-microsoft-com:office:office	Specifies whether a mouse click is detected on the fill of a shape. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
focus (Gradient Center)	Specifies the center starting position of a gradient. Values are in the range 100% to -100%. Default is 0. A value of 100% or -100% reverses the direction of the gradient (in effect swapping color and color2). A value of 50% changes the gradient so that color is at both ends and color2 is in the middle. A value of -50% changes the gradient so that color2 is at both ends and color is in the middle.

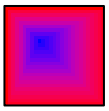
Attributes	Description

	[*Example*: The shape is filled with a horizontal gradient with red at both ends and blue in the middle:
	
	`<v:fill type="gradient"`
	`  color="red" color2="blue"`
	`  focus="50%">`
	`</v:fill>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

focusposition (Radial Gradient Center)	Specifies the position of the center rectangle of a radial gradient. The vector is a fraction of the width and height of the shape. The first is a percentage of the fill to the left edge; the second is a percentage of the fill to the top. Default is 0,0. To position a radial fill at the center of a shape, use a value of 50%,50%.
	
	[*Example*: The shape is filled with a rectangular gradient positioned in the top-left quadrant of the shape. The interior of the gradient is blue and the exterior is red:
	
	`<v:fill type="gradientradial"`
	`  color="red" color2="blue"`
	`  focusposition="25%,25%">`
	`</v:fill>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

focussize (Radial Gradient Size)	Specifies the size of the center rectangle of a radial gradient. The vector is a fraction of the width and height of the shape. The first is a percentage of the fill to the right edge; the second is a percentage of the fill to the bottom. Default is 0,0. A focussize value of 100%,100% and a focusposition of 0,0 makes color2 dominate the gradient completely. Small values of around 10%,10% are recommended for balanced gradients. [*Example*: The shape is filled with a rectangular gradient positioned in the top-left

Attributes	Description

	quadrant of the shape. The interior of the gradient is blue and the exterior is red. The red portion is wider on the bottom and right sides of the blue region. The pure blue region is 25% the width and 25% the height of the shape:
	
	`<v:fill type="gradientradial"`
	`  color="red" color2="blue"`
	`  focussize="25%,25%"`
	`  focusposition="25%,25%">`
	`</v:fill>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

href (Hyperlink Target)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the URL to the original image file. Used only if the picture has been linked and embedded. Default is no value.
	
	[*Example*:
	
	`<v:fill … o:href="myimage.gif" … >`
	`</v:fill>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

id (Relationship to Part) Namespace: …/officeDocument/2006/relationships	Specifies the relationship ID of the relationship to the image used for this fill. The specified relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/image or the document shall be considered non-conformant. [*Example*: The markup specifies the associated relationship part with relationship ID `rId10` contains the corresponding relationship information for the fill: `< … r:id="rId10" />` *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).
id (Unique	Specifies a unique identifier that can be used to reference a VML object.

Attributes	Description

Identifier)	Default is no value.
	
	[*Example*:
	
	`<v:shape … id="myShape" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

method (Gradient Fill Method)	Specifies the method used to generate the transition from color to color2 in a gradient fill. Default is `sigma`.
	
	[*Example*:
	
	`<v:fill type="gradient"`
	`  color="red" color2="blue" method="any">`
	`</v:fill>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_FillMethod simple type (§19.1.3.3).

on (Fill Toggle)	Specifies whether to fill the shape. Default is `true`. This attribute overrides the shape's fill attribute.
	
	[*Example*: The shape has a transparent fill:
	
	`<v:shape … fill="true" … >`
	`  <v:fill color="red" on="false">`
	`  </v:fill>`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

opacity (Primary Color Opacity)	Specifies the opacity of the primary fill color. Default is 1.0. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. For example, a value of "52429f"

Attributes	Description

	represents 52429/65536 or 0.8.
	
	[*Example*: The red color is 25% opaque:
	
	`<v:fill type="gradient" color="red"`
	`  color2="blue" opacity=".25">`
	`</v:fill>`
	
	  `opacity="1"`
	  `opacity=".25"`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

opacity2 (Secondary Color Opacity)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the opacity of the secondary fill color. Default is 1.0. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. [*Example*: For example, a value of "52429f" represents 52429/65536 or 0.8. *end example*]
	
	[*Example*: The blue color is 25% opaque:
	
	`<v:fill type="gradient" color="red"`
	`  color2="blue" o:opacity2=".25">`
	`</v:fill>`
	
	  `opacity2="1"`
	  `opacity2=".25"`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

origin (Fill Image Origin)	Specifies the position of the origin of a fill image as a point relative to the top left corner of the image. The vector is a fraction of the width and height of the image. Default is the center of the image. These numeric values can also be specified in 1/65536-ths if a trailing "f" is supplied. [*Example*: For example, a value of "52429f" represents 52429/65536 or 0.8. *end example*]

Attributes	Description

	[*Example*: The origin of the image is 25% to the right and 25% above the image's top left corner:
	
	`<v:fill type="tile" src="myimage.gif"`
	`  origin="0.25,-0.25">`
	`</v:fill>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

position (Fill Image Position)	Specifies the position of the origin of a fill image as a point within its containing shape. The vector is a fraction of the width and height of the shape. These numeric values can also be specified in 1/65536-ths if a trailing "f" is supplied. For example, a value of "52429f" represents 52429/65536 or 0.8.
	
	[*Example*: The origin of the image is positioned 25% to the right of the left edge of the shape and 25% down from the shape's top:
	
	`<v:fill type="tile" src="myimage.gif"`
	`  position="0.25,0.25">`
	`</v:fill>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

recolor (Recolor Fill as Picture)	Specifies that the fill uses an image. Default is `false`.
	
	[*Example*:
	
	`<v:fill r:id="rId4" o:title="MyPic" recolor="true"`
	`  type="frame">`
	`</v:fill>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

relid (Relationship to Part)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the relationship ID of the relationship to the image. The specified relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/image or the document shall be considered non-conformant.
	
	[*Example*: The markup specifies the associated relationship part with relationship ID `rId10` contains the corresponding relationship information:
	
	`<v:fill … o:relid="rId10" …>`
	`</v:fill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

rotate (Rotate Fill with Shape)	Specifies whether the fill is rotated with the shape. Default is `false`.
	
	[*Example*: The gradient is rotated with the shape:
	
	`<v:fill color2="white" focus="100%" rotate="true"`
	`type="gradient">`
	`</v:fill>`
	
	  `rotate="true"`
	
	  `rotate="false"`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

size (Fill Image Size)	Specifies the size of the fill image. Default is the native image pixel size.
	
	[*Example*: The image is reduced in size disproportionately:
	
	`<v:fill type="tile" src="myimage.gif"`
	`size="25pt,15pt">`
	`</v:fill>`

Attributes	Description
	 *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

src (Fill Image Source)	Specifies the URL specifying the fill image to use.
	
	[*Example*:
	
	`<v:fill … src="myimage.gif" … >`
	`</v:fill>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

title (Title)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the title of an embedded fill image. This is typically set to the comment property of the image, which is often blank.
	
	[*Example*:
	
	`<v:fill … o:title="alt text" … >`
	`</v:fill>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

type (Fill Type)	Specifies the kind of fill. Default is `solid`. Allowed values are: • `solid` • `gradient` • `gradientradial` • `tile` • `pattern` • `frame` [*Example*: Applied to a simple square using the following fill element, the three gradient types look like this:

Attributes	Description

	`<v:fill color="red" color2="blue"`
	`  type="solid">`
	`</v:fill>`
	
	  `type="solid"`
	  `type="gradient"`
	  `type="gradientradial"`
	
	Applied to a simple square using the following fill elements, the three image types look like this:
	
	<table><tr><td>  </td><td><v:fill src="myimage.gif"
	type="tile" size="50%,50%">
	</v:fill></td></tr><tr><td>  </td><td><v:fill src="myimage.gif"
	type="frame" size="50%,50%">
	</v:fill></td></tr><tr><td>  </td><td><v:fill src="myimage.gif"
	color="red" color2="blue"
	type="pattern">
	</v:fill></td></tr></table>
	end example]
	
	The possible values for this attribute are defined by the ST_FillType simple type (§19.1.3.4).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Fill) is located in §A.6.1. *end note*]

19.1.2.6 formulas (Set of Formulas)

This element defines a set of formulas whose calculated values are referenced by other attributes. Each formula is contained in a child f element (§19.1.2.4).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Formulas) is located in §A.6.1. *end note*]

19.1.2.7 group (Shape Group)

This element is used to collect shapes and groups so they can be positioned and transformed as a single unit. A group contains group, shapetype, shape, pre-defined shape - arc, curve, image, line, oval, polyline, rect, roundrect - and lock elements.

[*Example*: The following example defines a few basic parts of a flying saucer graphic. The group consists of five shapes. Each shape's position is determined within the coordinate space of the group, which is defined by the group's attributes.

```
<v:group id="saucer"
  style='position:relative;left:200;top:200;width:50;height:50'
  coordorigin="0,0" coordsize="6000,6000">
  <v:shape id="body"
    style='position:relative;left:234.75pt;top:208.875pt;
    width:235.25pt;height:128.875pt' coordsize="3765,2060"
    path="m1285,251l1126,469,580,1009,,1285,25,1412,93,1547,194,1673,
    1017,2026,2312,2060,3209,1756,3765,1388,3278,680,3059,319,2976,,
    1285,251,1285,251xe"
    fillcolor="#bcbcd6" stroked="f">
    <v:path arrowok="t"/>
  </v:shape>
  <v:shape id="canopy"
    style='position:relative;left:314.625pt;top:140.5pt;
    width:104pt;height:102pt' coordsize="1663,1633"
    path="m0,1355l177,1498,353,1582,840,1633,1378,1498,1663,1295,
    1545,456,1260,10,1025,,656,260,253,874,,1355,,1355xe"
    fillcolor="#99ebff" stroked="f">
    <v:path arrowok="t"/>
  </v:shape>
  <v:shape id="light1"
    style='position:relative;left:408.625pt;top:268.75pt;
    width:24.25pt;height:27.375pt' coordsize="388,437"
    path="m209,0l34,101,,302,125,437,329,327,388,152,209,,209,0xe"
    fillcolor="#fff27f" stroked="f">
    <v:path arrowok="t"/>
  </v:shape>
  <v:shape id="light2"
    style='position:relative;left:356.625pt;top:279.25pt;
    width:28.875pt;height:30pt' coordsize="462,479"
    path="m135,0l0,186,59,422,344,479,462,228,135,,135,0xe"
    fillcolor="#fff27f" stroked="f">
    <v:path arrowok="t"/>
  </v:shape>
```

```
    <v:shape id="light3"
      style='position:relative;left:302.625pt;top:274pt;
      width:23pt;height:23.625pt' coordsize="369,378"
      path="m0,59l226,,369,186,243,378,32,363,,59,,59xe"
      fillcolor="#fff27f" stroked="f">
      <v:path arrowok="t"/>
    </v:shape>
  </v:group>
```

end example]

Attributes	Description

allowincell (Allow in Table Cell)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can be placed in a table. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:allowincell="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

allowoverlap (Allow Shape Overlap)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can overlap another shape. Default is `true`. If `false`, the shape shifts left or right so as not to overlap another shape, similar to the behavior of the HTML float attribute.
	
	[*Example*:
	
	`<v:shape … o:allowoverlap="false" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

alt (Alternate Text)	Specifies alternative text describing the graphical object. This text should provide a brief description of the shape for use by accessibility tools. Default is no value. [*Example*: The alt text describes the basic shape: `<v:shape … fillcolor="red"`

Attributes	Description

	`alt="Red rectangle">`
	`</v:shape>`
	
	The alt text describes the contents of a shape displaying an image:
	
	`<v:shape … alt="Picture of a sunset">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderbottomcolor (Bottom Border Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the bottom border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderbottomcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderleftcolor (Border Left Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the left border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderleftcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderrightcolor (Border Right Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the right border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderrightcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

Attributes	Description

bordertopcolor (Border Top Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the top border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:bordertopcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

bullet (Graphical Bullet)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the shape is a graphical bullet. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:bullet="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

button (Button Behavior Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape exhibits button press behavior on click. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:button="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

class (CSS Reference)	Specifies a reference to the definition of a CSS style. Default is no value.
	
	[*Example*: The snippets below are equivalent:
	
	`…`
	`.narrowstyle {width:50;height:100}`
	`…`
	`<v:shape … class="narrowstyle"`
	`  style="top:1;left:1">`
	`</v:shape>`

Attributes	Description

	`<v:shape … style="top:1;left:1;`
	`  width:50;height:100">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

coordorigin (Coordinate Space Origin)	Specifies the coordinate of the top left corner of the shape's coordinate space. This determines the position of the (0,0) coordinate space origin within the shape's bounding box. Default is "0,0", which places the (0,0) origin at the top left corner of the bounding box.
	
	This affects shape properties that specify coordinate positions, such as the path attribute. Thus a path can be defined against a generic (0,0) origin and the coordorigin value translates the entire path within the shape's bounding space.
	
	[*Example*: The horizontal and vertical coordinate space ranges from -100 to +100 because the coordinate space (coordsize) is 200 by 200 and the top left coordinate is (-100,-100). The (0,0) origin lies at the center of the shape's bounding box, as evidenced by the position of the shape's path within the coordinate space:
	
	`<v:shape … coordsize="200,200"`
	`  coordorigin="-100,-100"`
	`  path="m 0,0 l 0,50, 50,50, 50,0 x e">`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

coordsize (Coordinate Space Size)	Specifies the size of the shape's coordinate space in coordinate units. Default is "1000,1000". The physical size of a coordinate unit length is determined by both the size of the coordinate space (coordsize) and the size of the shape (style width and height). The coordsize attribute defines the number of horizontal and vertical subdivisions into which the shape's bounding box is divided. The combination of coordsize and style width/height effective scales the shape anisotropically. [*Example*: The path is 50 units wide and tall, which is 25% of the size of the coordinate space:

Attributes	Description
	<v:shape … coordsize="200,200" coordorigin="-100,-100" path="m 0,0 l 0,50, 50,50, 50,0 x e"> </v:shape> *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
dgmlayout (Diagram Node Layout Identifier) Namespace: urn:schemas-microsoft-com:office:office	Specifies the kind of automatic layout to apply to the diagram node. This is only meaningful if the shape is a node in an organization chart, which is denoted by the `orgchart` value of the editas attribute of the group element. [*Example*: <v:shape … dgmlayout="1"> </v:shape> *end example*] The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).
dgmlayoutmru (Diagram Node Recent Layout Identifier) Namespace: urn:schemas-microsoft-com:office:office	Specifies the kind of automatic layout most recently used on the child elements of the diagram node. This is only meaningful if the shape is a node in anorganization chart, which is denoted by the `orgchart` value of the editas attribute of the group element. [*Example*: <v:shape … dgmlayout="1"> </v:shape> *end example*] The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).
dgmnodekind (Diagram Node Identifier) Namespace:	Specifies an optional, application-defined parameter that is intended to be used by the application to tag different types of nodes in a diagram. [*Example*:

Attributes	Description

urn:schemas-microsoft-com:office:office	`<v:shape … dgmnodekind="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

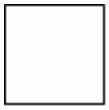
doubleclicknotify (Double-click Notification Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that an event message is sent when a shape is double-clicked. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:doubleclicknotify="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

editas (Group Diagram Type)	Specifies which diagram type the contained shapes represent. This is used in conjunction with the diagram element (§19.2.2.8). A value of `canvas` indicates that the group is a regular group of shapes and does not represent a diagram. Other values indicate that the diagram element and its children contain semantic information relevant to that type of diagram, which is represented by the shapes in the group.
	
	[*Example*:
	
	`<v:group … editas="orgchart">`
	`</v:group>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_EditAs simple type (§19.1.3.1).

fillcolor (Fill Color)	Specifies the color to use for the fill. Default is `white`. If the fill element (§19.1.2.5) is present, its color attribute takes precedence. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*: This shape is red if its fill is visible:
	
	`<v:shape … fillcolor="red" … >`
	`</v:shape>`
	
	This is equivalent to:

Attributes	Description

	`<v:shape … fillcolor="#ff0000" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

filled (Shape Fill Toggle)	Specifies whether the closed path is filled. Default is `true`. This attribute is overridden by the fill on attribute.
	
	[*Example*:
	
	`<v:shape … filled="f"`
	`  fillcolor="red" …>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hr (Horizontal Rule Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that a shape is a horizontal rule. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hr="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hralign (Horizontal Rule Alignment)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the alignment of a horizontal rule. Default is `left`.
	
	[*Example*:
	
	`<v:shape … o:hralign="center" … >`
	`</v:shape>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_HrAlign simple type (§19.2.3.16).

href (Hyperlink Target)	Specifies a hyperlink URL target for the shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … href="http://www.openxmlformats.org" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

hrnoshade (Horizontal Rule 3D Shading Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the horizontal rule does not have 3-D shading. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hrnoshade="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hrpct (Horizontal Rule Length Percentage)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the length of a horizontal rule as a percentage of page width. Default is 0.
	
	[*Example*:
	
	`<v:shape … o:hrpct="85" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema float datatype.

hrstd (Horizontal Rule Standard Display Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape is displayed as a standard horizontal rule. Only applies if hr is `true`. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hrstd="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type

Attributes	Description
	(§20.1.2.5).

id (Unique Identifier)	Specifies a unique identifier that can be used to reference a VML object.
	
	Default is no value.
	
	[*Example*:
	
	`<v:shape … id="myShape" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

insetmode (Text Inset Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the application calculates the internal text margin instead of using the inset attribute. Default is `custom`. This attribute is only meaningful for text boxes.
	
	[*Example*:
	
	`<v:shape … o:insetmode="auto" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_InsetMode simple type (§19.2.3.17).

oned (Shape Handle Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the extra handles of a shape are hidden. If `true`, hides all shape handles except the top left and bottom right; that is, the same handles that are used for a straight line segment. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oned="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

print (Print Toggle)	Specifies whether the shape is printed. Default is `true`.
	
	[*Example*:
	
	`<v:shape … print="false" … >`
	`</v:shape>`

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

regroupid (Regroup ID)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies a previous group for a shape. An ID number is used to identify groups of shapes that are no longer grouped. This allows shapes to be regrouped programmatically.
	
	[*Example*: The shape was part of a group identified by the ID 040754:
	
	`<v:shape … o:regroupid="040754" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

spid (Optional String) Namespace: urn:schemas-microsoft-com:office:office	Specifies an optional string that an application can use to Identify the particular shape. Default is no value. The possible values for this attribute are defined by the W3C XML Schema string datatype.

style (Shape Styling Properties)	Specifies the CSS2 styling properties of the shape. The CSS2 (Cascading Style Sheets, Level 2) specification, a Recommendation of the World Wide Web Consortium, is available here: http://www.w3.org/TR/REC-CSS2.
	
	This attribute uses a semi-colon delimited list of “name:value” pairs, the syntax defined by section 4.1.8 of the CSS2 specification for grouped declarations (without the surrounding braces). The following tables define the allowed properties and the VML treatment of each.
	
	[*Example*:
	`<v:shape … style=’position:absolute;width:100pt;height:50pt’ …`
	`>`
	`</v:shape>`
	end example]
	
	Property / **Description**
	`flip` — Specifies that the orientation of a shape is flipped. Default is no value. Allowed values are:
	• x - Flip along the y-axis, reversing the x-coordinates.

Attributes	Description	
		• y - Flip along the x-axis, reversing the y-coordinates. • xy - Flip along both the y- and x-axis. • yx - Flip along both the x- and y-axis.
	`height`	Specifies the height of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`left`	Specifies the position of the left of the containing block of the shape relative to the element left of it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-bottom`	Specifies the position of the bottom of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`margin-left`	Specifies the position of the left of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are:

Attributes	Description	
		• auto - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	margin-right	Specifies the position of the right of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • auto - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	margin-top	Specifies the position of the top of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • auto - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	mso-position-horizontal	Specifies the horizontal positioning data for objects in WordprocessingML documents. Default is absolute. Allowed values are: • absolute • left • center • right • inside • outside
	mso-position-horizontal-	Specifies relative horizontal position data for objects in WordprocessingML documents. This modifies the mso-

Attributes	Description	
	relative	position-horizontal property. Default is text. Allowed values are: • margin • page • text • char
	mso-position-vertical	Specifies the vertical positioning data for objects in WordprocessingML documents. Default is absolute. Allowed values are: • absolute • top • center • bottom • inside • outside
	mso-position-vertical-relative	Specifies relative vertical position data for objects in WordprocessingML documents. This modifies the mso-position-vertical property. Default is text. Allowed values are: • margin • page • text • line
	mso-wrap-distance-bottom	Specifies the distance from the bottom of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	mso-wrap-distance-left	Specifies the distance from the left side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	mso-wrap-distance-right	Specifies the distance from the right side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	mso-wrap-	Specifies the distance from the top of the shape to the text that

Attributes	Description
	`distance-top` — wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	`mso-wrap-edited` — Specifies whether the wrap coordinates were customized by the user. If the wrap coordinates are generated by an editor, this property is `true`; otherwise they were customized by a user. Default is `false`.
	`mso-wrap-style` — Specifies the wrapping mode for text in shapes in WordprocessingML documents. Default is `square`. Allowed values are: • `square` - Wraps text inside the shape in a square. • `none` - Text does not wrap.
	`position` — Specifies the kind of positioning used to place an element. Default is `static`. When the element is contained inside a group, this property shall be `absolute`. Allowed values are: • `static` - The element is positioned according to the normal flow of the page. The `top` and `left` properties are ignored. If the object is anchored inline, this value is used. • `absolute` - The element is positioned relative to the parent, using the `top` and `left` properties. • `relative` - The element is positioned according to the normal flow of the page, but the `top` and `left` properties are used. The overlap of overlapping elements is governed by the `z-index` property.
	`rotation` — Specifies the angle that a shape is rotated, in degrees. Default is 0. Positive angles are clockwise.
	`top` — Specifies the position of the top of the containing block of the shape relative to the element above it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.

Attributes	Description
	`visibility` — Specifies whether a shape is displayed. Only `inherit` and `hidden` are used; any other values are mapped to `inherit`. Default is `inherit`. Allowed values are: • `hidden` - The shape is not visible, but is still part of the flow of the objects in the browser. Mouse events are not processed. • `inherit` - The visibility state is inherited from the parent of the shape.
	`width` — Specifies the width of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`z-index` — Specifies the display order of overlapping shapes. Default is 0. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Uses the order that the shapes appear in the page, bottom to top. • <order>- A number that represents the stacking precedence. Shapes with higher numbers are placed on top of those with lower numbers. Negative numbers are allowed.
	The following properties are only used by the textbox element (§19.1.2.22):
	Property — **Description**
	`direction` — Specifies the direction of the text in the textbox. Default is `ltr`. This property is superceded by the `mso-direction-alt` property if that is specified. Allowed values are: • `ltr` - Text is displayed left-to-right. • `rtl` - Text is displayed right-to-left.
	`layout-flow` — Determines the flow of the text layout in a textbox. Default is `horizontal`. Allowed values are:

Attributes	Description
	• horizontal - Text is displayed horizontally. • vertical - Text is displayed vertically. • vertical-ideographic - Ideographic text is displayed vertically. • horizontal-ideographic - Ideographic text is displayed horizontally.
mso-direction-alt	Specifies an alternate direction for text in textboxes. Overrides the direction property. The only allowed value is context.
mso-fit-shape-to-text	Specifies whether the shape stretches to fit the text in the textbox. Default is false.
mso-fit-text-to-shape	Specifies whether the text stretches to fit the textbox. Default is false.
mso-layout-flow-alt	Specifies the alternate layout flow for text in textboxes. This property is used instead of layout-flow when the layout flow is from bottom to top for non-ideographic languages. Its only value is bottom-to-top.
mso-next-textbox	Specifies the ID of the next textbox in a series. Used to keep track of a set of linked textboxes. Default is no value.
mso-rotate	Specifies a specific rotation value for text in a textbox. Default is 0. Allowed values are: • 0 • 90 • 180 • -90
mso-text-scale	Specifies the scaling factor for fitting text to shapes. Default is 0. This property is only used if mso-fit-text-to-shape is true.
v-text-anchor	Specifies the vertical anchoring of text in a textbox. Default is top. The alignment of a text anchor only becomes evident if mso-fit-text-to-shape is false. This property is different from the vertical-align CSS property, which is used for ideographic languages. Allowed values are: • top • middle • bottom • top-center

Attributes	Description
	• `middle-center` • `bottom-center` • `top-baseline` • `bottom-baseline` • `top-center-baseline` • `bottom-center-baseline`

The following properties are only used by the textpath element (§19.1.2.23):

Property	Description
`font`	Specifies a compound value of font settings. Default is no value. The values are the same as those of the CSS font property. The order of definitions in the string is: font-style, font-variant, font-weight, font-size, line-height, font-family.
`font-family`	Specifies the family of the font. Default is no value. The values are the same as those of the CSS font-family property.
`font-size`	Specifies the size of the font. Default is no value. The font size is defined in points. The values are the same as those of the CSS font-size property.
`font-style`	Specifies the amount of slant for a font. Default is `normal`. The values are the same as those of the CSS font-style property. Allowed values are: • `normal` • `italic` • `oblique` - Treated the same as italic.
`font-variant`	Specifies the variant style of a font. Default is `normal`. The values are the same as those of the CSS font-variant property. Allowed values are: • `normal` • `small-caps`
`font-weight`	Specifies the thickness of the letters of the font. Default is `normal`. The values are the same as those of the CSS font-weight property. Allowed values are:

Value	Description
`normal` `lighter`	Treated as non-bold.

Attributes	Description
	<table><tr><td></td><td><table><tr><td>100 200 300 400</td><td></td></tr><tr><td>bold bolder 500 600 700 800 900</td><td>Treated as bold.</td></tr></table></td></tr><tr><td>mso-text-shadow</td><td>Specifies whether a shadow is applied to the text on a text path. Default is false.</td></tr><tr><td>text-decoration</td><td>Specifies the style of text decoration. Default is none. The values are the same as those of the CSS text-decoration property. Allowed values are: • none • underline • overline • line-through • blink</td></tr><tr><td>v-rotate-letters</td><td>Specifies whether the letters of the text are rotated counterclockwise by 90 degrees. Default is false.</td></tr><tr><td>v-same-letter-heights</td><td>Specifies whether all letters are the same height regardless of initial case. If true, the lowercase letters are stretched to the height of the uppercase letters. Default is false.</td></tr><tr><td>v-text-align</td><td>Specifies the alignment of text. Default is left. Allowed values are: • left • right • center • justify • letter-justify - Distributes the extra space between</td></tr></table>

Attributes	Description
	<table><tr><td></td><td>the letters. • stretch-justify - Stretches the letters to fill in the space.</td></tr><tr><td>v-text-kern</td><td>Specifies whether kerning is turned on. Default is false.</td></tr><tr><td>v-text-reverse</td><td>Specifies whether the layout order of rows is reversed. Default is false. This is used for vertical text layout.</td></tr><tr><td>v-text-spacing-mode</td><td>Specifies the mode for letter spacing. Default is tightening. This property determines whether space is removed between each letter (tightening) or added between each letter (tracking). The amount of letter spacing change is defined by the v-text-spacing property. Allowed values are: • tightening • tracking</td></tr><tr><td>v-text-spacing</td><td>Specifies the amount of spacing for text in 100ths of single line spacing. Default is 100.</td></tr></table> The line (§19.1.2.12), polyline (§19.1.2.15) and curve (§19.1.2.3) elements ignore the following properties: • top • left • width • height The following properties are not inherited by an element that references a shapetype element (§19.1.2.20) via the id attribute: • flip • height • left • margin-left • margin-top • position • rotation • top • visibility • width • z-index The possible values for this attribute are defined by the W3C XML Schema string datatype.

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>tablelimits (Table Row Height Limits)

Namespace: urn:schemas-microsoft-com:office:office</td><td>Specifies a list of minimum height values for each row in a table. Default is no value.

Used by PresentationML for native tables. This attribute is only useful when the table is made up of shapes that are grouped. When text is added to table cells, the row height can increase. The tablelimits attribute stores the original row height so that if text is deleted, the row height does not fall below the original value.

[Example:

<pre><v:shape … o:tablelimits="30pt 20pt" … >
</v:shape></pre>
end example]

The possible values for this attribute are defined by the W3C XML Schema string datatype.</td></tr>
<tr><td>tableproperties (Table Properties)

Namespace: urn:schemas-microsoft-com:office:office</td><td>Specifies a bitmask, represented as an integer, that determines table properties. Only the first three bits of this integer are used. Default is 0.

Used by PresentationML for native tables. This attribute is only useful when the table is made up of shapes that are grouped. Allowed values are:

<table><tr><th>Bit</th><th>Description</th></tr><tr><td>1</td><td>Set if the group of shapes is a table.</td></tr><tr><td>2</td><td>Set if the shape is a placeholder.</td></tr><tr><td>3</td><td>Set if the table text is bi-directional.</td></tr></table>
[Example: Decimal 3 means that bits 1 and 2 are set.

<pre><v:shape … o:tableproperties="3" … >
</v:shape></pre>
end example]

The possible values for this attribute are defined by the W3C XML Schema string datatype.</td></tr>
<tr><td>target (Hyperlink Display Target)</td><td>Specifies a frame or window that a URL is displayed in. Default is no value. Allowed values are:

<table><tr><th>Value</th><th>Description</th></tr><tr><td><targetname></td><td>String containing the name of the frame or window in which to load the document.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>_blank</td><td>Specifies that the linked document is loaded into a new blank window. This window is not named.</td></tr>
<tr><td>_media</td><td>Specifies that the linked document is loaded into the browser’s multimedia pane.</td></tr>
<tr><td>_parent</td><td>Specifies that the linked document is loaded into the immediate parent of the document containing the link.</td></tr>
<tr><td>_search</td><td>Specifies that the linked document is loaded into the browser's search pane.</td></tr>
<tr><td>_self</td><td>Specifies that the linked document is loaded into the window in which the link was clicked (the active window).</td></tr>
<tr><td>_top</td><td>Specifies that the linked document is loaded into the topmost window.</td></tr>
</table>
[Example:
<pre>
<v:shape …
  href="http://www.openxmlformats.org"
  target="_self" … >
</v:shape>
</pre>
end example]
The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>title (Shape Title)</td><td>
Specifies the text displayed when the mouse pointer moves over the shape. Default is no value.
[Example:
<pre>
<v:shape … title="tooltip" … >
</v:shape>
</pre>
end example]
The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>userdrawn (Exists In Master Slide)

Namespace: urn:schemas-microsoft-com:office:office</td><td>
Specifies whether the user has added the shape to a master slide. Default is <code>false</code>. Used by PresentationML.
[Example:
<pre>
<v:shape … o:userdrawn="true" … >
</v:shape>
</pre>
</td></tr>
</table>

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

userhidden (Hide Script Anchors)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a script anchor is hidden. Default is `false`. If `true`, script anchors stay hidden even if the shape is otherwise visible. A script anchor is the visual representation of a script that when displayed in an application.
	
	[*Example*:
	
	`<v:shape … o:userhidden="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

wrapcoords (Shape Bounding Polygon)	Specifies the bounding polygon that surrounds a shape. This is specified using a comma-delimited list of x and y coordinates: "x1,y1,x2,y2,x3,y3,…" This is used when text is tightly wrapped around a shape. Default is no value until the mso-wrap-mode style attribute is set to `tight` or `through`.
	
	[*Example*:
	
	`<v:shape …`
	`wrapcoords="0,0 0,200, 200,200, 200,0" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Group) is located in §A.6.1. *end note*]

19.1.2.8 h (Shape Handle)

This element defines a single handle, which is a user interface element tied to one or two adj values. Moving the handle changes its linked adj values, which in turn changes formulas and attributes that depend on them. The handle is optionally constrained vertically or horizontally. The linked adj values store the position of the handle in the shape's coordinate space.

[*Example*: The example below defines a simple kite shape with a resizable width:

```
<v:shape coordsize="200,200" coordorigin="-100,-100" adj="100"
style="width:50;height:50;position:relative"
 path="m @1,-50 l 0,-200 @0,-50 0,200 x e">
 <v:formulas>
  <v:f eqn="val #0"/>
  <v:f eqn="sum 0 0 @0"/>
 </v:formulas>
 <v:handles>
  <v:h position="#0,0"/>
 </v:handles>
</v:shape>
```

end example]

Attributes	Description

invx (Invert Handle's X Position)	Specifies whether the x position of the handle should be inverted according to: $x_{new} = coordorigin_x + coordsize_x - x_{old}$
	
	Default is `false`.
	
	[*Example*:
	
	`<v:handles>`
	`  <v:h … invx="true" … />`
	`</v:handles>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

invy (Invert Handle's Y Position)	Specifies whether the y position of the handle should be inverted according to: $y_{new} = coordorigin_y + coordsize_y - y_{old}$
	
	Default is `false`.
	
	[*Example*:
	
	`<v:handles>`
	`  <v:h … invy="true" … />`
	`</v:handles>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

map (Handle	Specifies how the x and y positions of the handle are mapped from the coordsize range

Attributes	Description

Coordinate Mapping)	into the specified range. Default is "0,1000".
	
	[*Example*:
	
	`<v:handles>`
	`  <v:h … map="-1000,1000" … />`
	`</v:handles>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

polar (Handle Polar Center)	Specifies the center position of a handle that uses polar coordinates. If specified, the position attribute is assumed to contain radius and angle values. If omitted, the position attribute is assumed to contain x and y positions. Default is no value.
	
	[*Example*:
	
	`<v:handles>`
	`  <v:h … polar="0,0" … />`
	`</v:handles>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

position (Handle Position)	Specifies the x and y position of the handle. If the polar attribute is present, defines the handle position using radius and angle values. Default is "0,0".
	
	Each value in the vector is one of the following:
	• constant
	• formula (e.g., @2)
	• adj value (e.g., #2)
	• `center`
	• `topleft`
	• `bottomright`
	
	Each of the above except for an adj value reference fixes the handle position for that dimension. Specifying an adj value allows the handle to move in that dimension and the handle position for that dimension is stored in the adj value.
	
	[*Example*: The handle's x position is fixed but it is free to move in the y dimension:
	
	`<v:handles>`
	`  <v:h … position="topleft,#2" … />`

Attributes	Description
	`</v:handles>` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

radiusrange (Handle Polar Radius Range)	Specifies a range of minimum and maximum values that constrain the radius of a handle using polar coordinates. Default is "0,0". Each value is either a constant or a formula reference. Omitting a value leaves that bound unconstrained.
	
	[*Example*: The polar handle can only be moved within a radius range of 25 to 50.
	
	`<v:handles>`
	`  <v:h … radiusrange="25,50" … />`
	`</v:handles>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

switch (Handle Inversion Toggle)	Specifies whether the x and y dimensions of the handle are switched when the shape is taller than it is wide. Default is `false`. This is useful for shapes with limo stretch behavior.
	
	[*Example*:
	
	`<v:handles>`
	`  <v:h … switch="true" … />`
	`</v:handles>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

xrange (Handle X Position Range)	Specifies a range of minimum and maximum values that constrain the x position of a handle. Default is "0,0". Each value is either a constant or a formula reference. Omitting a value leaves that bound unconstrained.
	
	[*Example*: The handle's x position has a maximum bound of 500 and no minimum bound:
	
	`<v:handles>`
	`  <v:h … xrange=",500" … />`
	`</v:handles>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

yrange (Handle Y Position Range)	Specifies a range of minimum and maximum values that constrain the y position of a handle. Default is "0,0". Each value is either a constant or a formula reference. Omitting a value leaves that bound unconstrained.
	
	[*Example*: The handle's y position has a minimum bound of -500 and no maximum bound:
	
	`<v:handles>`
	`  <v:h … yrange="-500," … />`
	`</v:handles>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_H) is located in §A.6.1. *end note*]

19.1.2.9 handles (Set of Handles)

This element defines a set of user interface elements which can vary a shape's adj values. All dependent formulas and attributes are recalculated. Each handle is defined by a child h element.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Handles) is located in §A.6.1. *end note*]

19.1.2.10 image (Image File)

This element is used to draw an image that has been loaded from an external source. There is an implied rectangle that is the same size as the image. Any stroke or fill is applied to this implied rectangle. The stroke is drawn on top of the image. The fill is behind the image and therefore only visible through transparent areas of the image. Image transparency is either encoded in the file or defined via a color value using the chromakey attribute. Unlike the imagedata element (§19.1.2.11), the image element does not have a parent element.

[*Example*:

```
<v:image src="myimage.gif"
  style="position:relative;top:1;left:1;width:50;height:45"
  cropbottom="10%" gamma="0.5" gain="2">
</v:image>
```

Original image: 

Modified image: 

end example]

Attributes	Description

allowincell (Allow in Table Cell)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can be placed in a table. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:allowincell="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

allowoverlap (Allow Shape Overlap)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can overlap another shape. Default is `true`. If `false`, the shape shifts left or right so as not to overlap another shape, similar to the behavior of the HTML float attribute.
	
	[*Example*:
	
	`<v:shape … o:allowoverlap="false" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

alt (Alternate Text)	Specifies alternative text describing the graphical object. This text should provide a brief description of the shape for use by accessibility tools. Default is no value.
	
	[*Example*: The alt text describes the basic shape:
	
	`<v:shape … fillcolor="red"`
	`  alt="Red rectangle">`
	`</v:shape>`
	
	The alt text describes the contents of a shape displaying an image:
	
	`<v:shape … alt="Picture of a sunset">`

Attributes	Description
	`</v:shape>` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

bilevel (Image Bilevel Toggle)	Specifies that all colors in the picture shall be converted to either 0 or full intensity component values. This converts a color bitmap to 8 colors and a grayscale bitmap to black and white. Default is `false`.
	
	[*Example*:
	
	`<v:image … bilevel="true" …>`
	`</v:image>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

blacklevel (Image Brightness)	Specifies the image brightness. Default is 0.
	
	[*Example*:
	
	`<v:image … blacklevel="0.1" …>`
	`</v:image>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderbottomcolor (Bottom Border Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the bottom border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderbottomcolor="red" … >`
	`</v:shape>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderleftcolor (Border Left Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the left border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderleftcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderrightcolor (Border Right Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the right border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderrightcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

bordertopcolor (Border Top Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the top border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:bordertopcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

bullet (Graphical Bullet)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the shape is a graphical bullet. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:bullet="true" … >`
	`</v:shape>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

button (Button Behavior Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape exhibits button press behavior on click. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:button="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

bwmode (Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies how a shape renders for black-and-white output devices. When a shape is printed on a black-and-white printer or displayed in a black-and-white view in an application, several options are possible. Default is `auto`, which uses o:bwnormal for normal black-and-white rendering and o:bwpure for pure black-and-white rendering.
	
	bwnormal and bwpure are subordinate to bwmode. If bwmode is "auto" then the value for bwnormal or bwpure is used depending on what the output format is. An application can define for itself what, if any, difference there is between normal B&W and pure B&W. [*Example*: Normal B&W might allow greyscale and pure B&W might not. *end example*]
	[*Example*: This shape renders in grayscale in a black-and-white environment:
	
	`<v:shape … o:bwmode="grayscale" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

bwnormal (Normal Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the black-and-white mode for normal black-and-white output devices. Default is `auto`.
	
	[*Example*: This shape renders in a pale grayscale in a normal black-and-white environment:
	
	`<v:shape …  o:bwmode="auto" o:bwnormal="lightgrayscale" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

Attributes	Description

bwpure (Pure Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the black-and-white mode for pure black-and-white output devices. Default is auto.
	
	[*Example*: This shape renders in high contrast when in a pure black-and-white environment:
	
	`<v:shape … o:bwmode="auto" o:bwpure="highcontrast" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

chromakey (Image Transparency Color)	Specifies a color value that is transparent and show anything behind the shape. Default is no value.
	
	[*Example*:
	
	`<v:image … chromakey="white" …>`
	`</v:image>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

class (CSS Reference)	Specifies a reference to the definition of a CSS style. Default is no value.
	
	[*Example*: The snippets below are equivalent:
	
	`…`
	`.narrowstyle {width:50;height:100}`
	`…`
	`<v:shape … class="narrowstyle"`
	`  style="top:1;left:1">`
	`</v:shape>`
	
	`<v:shape … style="top:1;left:1;`
	`  width:50;height:100">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

Attributes	Description

clip (Clipping Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the clipping region is active. This is used in conjunction with the clippath (§19.2.2.3) element to create a clipping region.
	
	[*Example*:
	
	`<v:shape … o:clip="true">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

cliptowrap (Clip to Wrapping Polygon)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the clipping region for the shape aligns with the wrapping polygon that tightly surrounds the entire shape (essentially, that the shape shall not be drawn beyond its wrapping polygon's extents – if it does, the shape shall be clipped). Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:cliptowrap="true">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

connectortype (Shape Connector Type)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of connector used for joining shapes. Default is `straight`.
	
	[*Example*:
	
	`<v:shape … o:connectortype="elbow" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ConnectorType simple type (§19.2.3.7).

coordorigin (Coordinate Space Origin)	Specifies the coordinate of the top left corner of the shape's coordinate space. This determines the position of the (0,0) coordinate space origin within the shape's bounding box. Default is "0,0", which places the (0,0) origin at the top left corner of the bounding box. This affects shape properties that specify coordinate positions, such as the path attribute. Thus a path can be defined against a generic (0,0) origin and the coordorigin value translates the entire path within the shape's bounding space.

Attributes	Description

	[*Example*: The horizontal and vertical coordinate space ranges from -100 to +100 because the coordinate space (coordsize) is 200 by 200 and the top left coordinate is (-100,-100). The (0,0) origin lies at the center of the shape's bounding box, as evidenced by the position of the shape's path within the coordinate space:
	
	`<v:shape … coordsize="200,200"`
	`  coordorigin="-100,-100"`
	`  path="m 0,0 l 0,50, 50,50, 50,0 x e">`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

coordsize (Coordinate Space Size)	Specifies the size of the shape's coordinate space in coordinate units. Default is "1000,1000".
	
	The physical size of a coordinate unit length is determined by both the size of the coordinate space (coordsize) and the size of the shape (style width and height). The coordsize attribute defines the number of horizontal and vertical subdivisions into which the shape's bounding box is divided. The combination of coordsize and style width/height effective scales the shape anisotropically.
	
	[*Example*: The path is 50 units wide and tall, which is 25% of the size of the coordinate space:
	
	`<v:shape … coordsize="200,200"`
	`  coordorigin="-100,-100"`
	`  path="m 0,0 l 0,50, 50,50, 50,0 x e">`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

cropbottom (Image Bottom Crop)	Specifies the how much to crop the image from the bottom up as a fraction of picture size. Default is 0. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. For example, a value of "52429f" represents 52429/65536 or 0.8.

Attributes	Description

	[*Example*:
	
	`<v:image … cropbottom="10%" …>`
	`</v:image>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

cropleft (Image Left Crop)	Specifies how much to crop the image from the left in as a fraction of picture size. Default is 0. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. For example, a value of "52429f" represents 52429/65536 or 0.8.
	
	[*Example*:
	
	`<v:image … cropleft="10%" …>`
	`</v:image>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

cropright (Image Right Crop)	Specifies how much to crop the image from the right in as a fraction of picture size. Default is 0. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. [*Example*: For example, a value of "52429f" represents 52429/65536 or 0.8. *end example*]
	
	[*Example*:
	
	`<v:image … cropright="10%" …>`
	`</v:image>`
	
	 
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

croptop (Image Top Crop)	Specifies how much to crop the image from the top down as a fraction of picture size. Default is 0. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. [*Example:* For example, a value of "52429f" represents 52429/65536 or 0.8. *end example*]
	
	[*Example*:
	
	`<v:image … croptop="10%" …>`
	`</v:image>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

dgmlayout (Diagram Node Layout Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of automatic layout to apply to the diagram node. This is only meaningful if the shape is a node in an organization chart, which is denoted by the `orgchart` value of the editas attribute of the group element.
	[*Example*:
	
	`<v:shape … dgmlayout="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).

dgmlayoutmru (Diagram Node Recent Layout Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of automatic layout most recently used on the child elements of the diagram node. This is only meaningful if the shape is a node in anorganization chart, which is denoted by the `orgchart` value of the editas attribute of the group element.
	
	[*Example*:
	
	`<v:shape … dgmlayout="1">`
	`</v:shape>`
	
	end example]

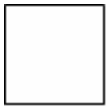
Attributes	Description
	The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).

dgmnodekind (Diagram Node Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies an optional, application-defined parameter that is intended to be used by the application to tag different types of nodes in a diagram.
	
	[*Example*:
	
	`<v:shape … dgmnodekind="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

doubleclicknotify (Double-click Notification Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that an event message is sent when a shape is double-clicked. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:doubleclicknotify="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

fillcolor (Fill Color)	Specifies the color to use for the fill. Default is `white`. If the fill element (§19.1.2.5) is present, its color attribute takes precedence. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*: This shape is red if its fill is visible:
	
	`<v:shape … fillcolor="red" … >`
	`</v:shape>`
	
	This is equivalent to:
	
	`<v:shape … fillcolor="#ff0000" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

Attributes	Description

filled (Shape Fill Toggle)	Specifies whether the closed path is filled. Default is `true`. This attribute is overridden by the fill on attribute.
	
	[*Example*:
	
	`<v:shape … filled="f"`
	`  fillcolor="red" …>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

forcedash (Force Dashed Outline)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a dashed outline is used to draw a shape when a shape has no line or fill. Default is `false`.
	
	Used by PresentationML placeholders to draw a dashed outline when there is no line and no fill for a shape.
	
	[*Example*:
	
	`<v:shape … o:forcedash="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

gain (Image Intensity)	Specifies an adjustment for the intensity of all colors. Essentially sets how bright white is. Default is 1.
	
	[*Example*:
	
	`<v:image … gain="0.5" …>`
	`</v:image>`
	
	 
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

gamma (Image Gamma Correction)	Specifies the gamma correction. Default is 1.
	
	Gamma correction is a factor by which the intended target display gamma differs from the sRGB profile. It can be used to correct for images not prepared for sRGB displays and to adjust overall image contrast. Decreasing it below 1 gives a higher contrast image.
	
	[*Example*:
	
	`<v:image … gamma="0.5" …>`
	`</v:image>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

grayscale (Image Grayscale Toggle)	Specifies to display the image in grayscale. Default is `false`.
	
	[*Example*:
	
	`<v:image … gamma="0.5" …>`
	`</v:image>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hr (Horizontal Rule Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that a shape is a horizontal rule. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hr="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type

Attributes	Description
	(§20.1.2.5).

hralign (Horizontal Rule Alignment)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the alignment of a horizontal rule. Default is `left`.
	
	[*Example*:
	
	`<v:shape … o:hralign="center" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_HrAlign simple type (§19.2.3.16).

href (Hyperlink Target)	Specifies a hyperlink URL target for the shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … href="http://www.openxmlformats.org" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

hrnoshade (Horizontal Rule 3D Shading Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the horizontal rule does not have 3-D shading. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hrnoshade="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hrpct (Horizontal Rule Length Percentage)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the length of a horizontal rule as a percentage of page width. Default is 0.
	
	[*Example*:
	
	`<v:shape … o:hrpct="85" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema float datatype.

hrstd (Horizontal	Specifies whether a shape is displayed as a standard horizontal rule. Only applies if hr is

Attributes	Description

Rule Standard Display Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	`true`. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hrstd="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

id (Unique Identifier)	Specifies a unique identifier that can be used to reference a VML object.
	
	Default is no value.
	
	[*Example*:
	
	`<v:shape … id="myShape" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

insetmode (Text Inset Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the application calculates the internal text margin instead of using the inset attribute. Default is `custom`. This attribute is only meaningful for text boxes.
	
	[*Example*:
	
	`<v:shape … o:insetmode="auto" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_InsetMode simple type (§19.2.3.17).

insetpen (Inset Border From Path)	Specifies that the border shall be displayed inside of the path defining the shape, rather than along the path (the default border placement), or outside of the path as might be done with an image.
	
	[*Example*:
	
	`<v:shape … insetpen="true" … >`
	`</v:shape>`

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

ole (Embedded Object Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the shape is an embedded object. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:ole="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

oleicon (Embedded Object Icon Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether an embedded object is displayed as an icon. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oleicon="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

oned (Shape Handle Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the extra handles of a shape are hidden. If `true`, hides all shape handles except the top left and bottom right; that is, the same handles that are used for a straight line segment. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oned="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

opacity (Fill Color Opacity)	Specifies the opacity of the primary fill color. Default is 1.0. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. [*Example*: A value of "52429f" represents 52429/65536 or 0.8. *end example*] [*Example*: The red color is 25% opaque:

Attributes	Description

	`<v:fill type="gradient" color="red"`
	`  color2="blue" opacity=".25">`
	`</v:fill>`
	
	  `opacity="1"`
	  `opacity=".25"`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

preferrelative (Relative Resize Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the original size of an object is saved after reformatting. Default is `false`. If `true`, the original size of the object is stored and all resizing is based on a percentage of that original size. Otherwise, each resizing resets the scale to 100%.
	
	[*Example*:
	
	`<v:shape … o:preferrelative="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

print (Print Toggle)	Specifies whether the shape is printed. Default is `true`.
	
	[*Example*:
	
	`<v:shape … print="false" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

regroupid (Regroup ID) Namespace: urn:schemas-microsoft-	Specifies a previous group for a shape. An ID number is used to identify groups of shapes that are no longer grouped. This allows shapes to be regrouped programmatically. [*Example*: The shape was part of a group identified by the ID 040754: `<v:shape … o:regroupid="040754" … >`

Attributes	Description
com:office:office	`</v:shape>` *end example*] The possible values for this attribute are defined by the W3C XML Schema integer datatype.
spid (Optional String) Namespace: urn:schemas-microsoft-com:office:office	Specifies an optional string that an application can use to Identify the particular shape. Default is no value. The possible values for this attribute are defined by the W3C XML Schema string datatype.
spt (Optional Number) Namespace: urn:schemas-microsoft-com:office:office	Specifies an optional number that an application can use to associate the particular shape with a defined shape type. Default is 0. The possible values for this attribute are defined by the W3C XML Schema float datatype.

src (Image Source)	Specifies the URL of the image to use.
	
	[*Example*:
	
	`<v:image … src="myimage.gif" …>`
	`</v:image>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

strokecolor (Shape Stroke Color)	Specifies the primary color of the brush to use to stroke the path of the shape. Default is `black`. The color attribute of the stroke element (§19.1.2.21) overrides this. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*:
	
	`<v:shape … strokecolor="red" …>`
	`</v:shape>`

Attributes	Description
	 *end example*] The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

stroked (Shape Stroke Toggle)	Specifies whether the path defining the shape is stroked with a solid line. The stroke element (§19.1.2.21) defines other strokes. The on attribute of the stroke element overrides this attribute. Default is true.
	
	[*Example*:
	
	`<v:shape … fillcolor="red"`
	`  stroked="false" strokecolor="blue"…>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

strokeweight (Shape Stroke Weight)	Specifies the width of the brush to use to stroke the path. Default is 1 point. If a number is given without units, the emu is used. The weight attribute of the stroke element (§19.1.2.21) overrides this attribute.
	
	[*Example*:
	
	`<v:shape … strokeweight="3pt" … >`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

style (Shape Styling	Specifies the CSS2 styling properties of the shape. The CSS2 (Cascading Style Sheets, Level

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>Properties)</td><td>2) specification, a Recommendation of the World Wide Web Consortium, is available here: http://www.w3.org/TR/REC-CSS2.

This attribute uses a semi-colon delimited list of “name:value” pairs, the syntax defined by section 4.1.8 of the CSS2 specification for grouped declarations (without the surrounding braces). The following tables define the allowed properties and the VML treatment of each.

[Example:
<pre><v:shape … style=’position:absolute;width:100pt;height:50pt’ …
>
</v:shape></pre>end example]

<table>
<tr><th>Property</th><th>Description</th></tr>
<tr><td>flip</td><td>Specifies that the orientation of a shape is flipped. Default is no value. Allowed values are:

• x - Flip along the y-axis, reversing the x-coordinates.
• y - Flip along the x-axis, reversing the y-coordinates.
• xy - Flip along both the y- and x-axis.
• yx - Flip along both the x- and y-axis.</td></tr>
<tr><td>height</td><td>Specifies the height of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are:

• auto - Default position of an element in the flow of the page.
• <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed.
• <percentage>- Value expressed as a percentage of the parent object's height.</td></tr>
<tr><td>left</td><td>Specifies the position of the left of the containing block of the shape relative to the element left of it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are:

• auto - Default position of an element in the flow of the page.
• <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed.</td></tr>
</table></td></tr>
</table>

Attributes	Description	
		• <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-bottom`	Specifies the position of the bottom of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`margin-left`	Specifies the position of the left of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-right`	Specifies the position of the right of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-top`	Specifies the position of the top of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the

Attributes	Description	
		page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`mso-position-horizontal`	Specifies the horizontal positioning data for objects in WordprocessingML documents. Default is `absolute`. Allowed values are: • `absolute` • `left` • `center` • `right` • `inside` • `outside`
	`mso-position-horizontal-relative`	Specifies relative horizontal position data for objects in WordprocessingML documents. This modifies the `mso-position-horizontal` property. Default is `text`. Allowed values are: • `margin` • `page` • `text` • `char`
	`mso-position-vertical`	Specifies the vertical positioning data for objects in WordprocessingML documents. Default is `absolute`. Allowed values are: • `absolute` • `top` • `center` • `bottom` • `inside` • `outside`
	`mso-position-vertical-relative`	Specifies relative vertical position data for objects in WordprocessingML documents. This modifies the `mso-position-vertical` property. Default is `text`. Allowed values are: • `margin`

Attributes	Description
	• page • text • line
mso-wrap-distance-bottom	Specifies the distance from the bottom of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
mso-wrap-distance-left	Specifies the distance from the left side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
mso-wrap-distance-right	Specifies the distance from the right side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
mso-wrap-distance-top	Specifies the distance from the top of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
mso-wrap-edited	Specifies whether the wrap coordinates were customized by the user. If the wrap coordinates are generated by an editor, this property is true; otherwise they were customized by a user. Default is false.
mso-wrap-style	Specifies the wrapping mode for text in shapes in WordprocessingML documents. Default is square. Allowed values are: • square - Wraps text inside the shape in a square. • none - Text does not wrap.
position	Specifies the kind of positioning used to place an element. Default is static. When the element is contained inside a group, this property shall be absolute. Allowed values are: • static - The element is positioned according to the normal flow of the page. The top and left properties are ignored. If the object is anchored inline, this value is used. • absolute - The element is positioned relative to the parent, using the top and left properties.

Attributes	Description
	• `relative` - The element is positioned according to the normal flow of the page, but the `top` and `left` properties are used. The overlap of overlapping elements is governed by the `z-index` property.
`rotation`	Specifies the angle that a shape is rotated, in degrees. Default is 0. Positive angles are clockwise.
`top`	Specifies the position of the top of the containing block of the shape relative to the element above it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
`visibility`	Specifies whether a shape is displayed. Only `inherit` and `hidden` are used; any other values are mapped to `inherit`. Default is `inherit`. Allowed values are: • `hidden` - The shape is not visible, but is still part of the flow of the objects in the browser. Mouse events are not processed. • `inherit` - The visibility state is inherited from the parent of the shape.
`width`	Specifies the width of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
`z-index`	Specifies the display order of overlapping shapes. Default is 0. This property shall not be used for shapes anchored inline. Allowed values are:

Attributes	Description
	<table><tr><td></td><td>• auto - Uses the order that the shapes appear in the page, bottom to top. • <order>- A number that represents the stacking precedence. Shapes with higher numbers are placed on top of those with lower numbers. Negative numbers are allowed.</td></tr></table> The following properties are only used by the textbox element (§19.1.2.22): <table><tr><th>Property</th><th>Description</th></tr><tr><td>direction</td><td>Specifies the direction of the text in the textbox. Default is ltr. This property is superceded by the mso-direction-alt property if that is specified. Allowed values are: • ltr - Text is displayed left-to-right. • rtl - Text is displayed right-to-left.</td></tr><tr><td>layout-flow</td><td>Determines the flow of the text layout in a textbox. Default is horizontal. Allowed values are: • horizontal - Text is displayed horizontally. • vertical - Text is displayed vertically. • vertical-ideographic - Ideographic text is displayed vertically. • horizontal-ideographic - Ideographic text is displayed horizontally.</td></tr><tr><td>mso-direction-alt</td><td>Specifies an alternate direction for text in textboxes. Overrides the direction property. The only allowed value is context.</td></tr><tr><td>mso-fit-shape-to-text</td><td>Specifies whether the shape stretches to fit the text in the textbox. Default is false.</td></tr><tr><td>mso-fit-text-to-shape</td><td>Specifies whether the text stretches to fit the textbox. Default is false.</td></tr><tr><td>mso-layout-flow-alt</td><td>Specifies the alternate layout flow for text in textboxes. This property is used instead of layout-flow when the layout flow is from bottom to top for non-ideographic languages. Its only value is bottom-to-top.</td></tr><tr><td>mso-next-textbox</td><td>Specifies the ID of the next textbox in a series. Used to keep track of a set of linked textboxes. Default is no value.</td></tr></table>

Attributes	Description

mso-rotate	Specifies a specific rotation value for text in a textbox. Default is 0. Allowed values are: • 0 • 90 • 180 • -90
mso-text-scale	Specifies the scaling factor for fitting text to shapes. Default is 0. This property is only used if mso-fit-text-to-shape is true.
v-text-anchor	Specifies the vertical anchoring of text in a textbox. Default is top. The alignment of a text anchor only becomes evident if mso-fit-text-to-shape is false. This property is different from the vertical-align CSS property, which is used for ideographic languages. Allowed values are: • top • middle • bottom • top-center • middle-center • bottom-center • top-baseline • bottom-baseline • top-center-baseline • bottom-center-baseline

The following properties are only used by the textpath element (§19.1.2.23):

Property	Description
font	Specifies a compound value of font settings. Default is no value. The values are the same as those of the CSS font property. The order of definitions in the string is: font-style, font-variant, font-weight, font-size, line-height, font-family.
font-family	Specifies the family of the font. Default is no value. The values are the same as those of the CSS font-family property.
font-size	Specifies the size of the font. Default is no value. The font size is defined in points. The values are the same as those of the CSS font-size property.
font-style	Specifies the amount of slant for a font. Default is normal. The values are the same as those of the CSS font-style property. Allowed values are:

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>

- `normal`
- `italic`
- `oblique` - Treated the same as italic.

</td></tr>
<tr><td>`font-variant`</td><td>Specifies the variant style of a font. Default is `normal`. The values are the same as those of the CSS font-variant property. Allowed values are:

- `normal`
- `small-caps`

</td></tr>
<tr><td>`font-weight`</td><td>Specifies the thickness of the letters of the font. Default is `normal`. The values are the same as those of the CSS font-weight property. Allowed values are:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>`normal`
`lighter`
`100`
`200`
`300`
`400`</td><td>Treated as non-bold.</td></tr>
<tr><td>`bold`
`bolder`
`500`
`600`
`700`
`800`
`900`</td><td>Treated as bold.</td></tr>
</table>

</td></tr>
<tr><td>`mso-text-shadow`</td><td>Specifies whether a shadow is applied to the text on a text path. Default is `false`.</td></tr>
<tr><td>`text-decoration`</td><td>Specifies the style of text decoration. Default is `none`. The values are the same as those of the CSS text-decoration property. Allowed values are:

- `none`

</td></tr>
</table>
</td></tr>
</table>

Attributes	Description
	• underline • overline • line-through • blink
v-rotate-letters	Specifies whether the letters of the text are rotated counterclockwise by 90 degrees. Default is false.
v-same-letter-heights	Specifies whether all letters are the same height regardless of initial case. If true, the lowercase letters are stretched to the height of the uppercase letters. Default is false.
v-text-align	Specifies the alignment of text. Default is left. Allowed values are: • left • right • center • justify • letter-justify - Distributes the extra space between the letters. • stretch-justify - Stretches the letters to fill in the space.
v-text-kern	Specifies whether kerning is turned on. Default is false.
v-text-reverse	Specifies whether the layout order of rows is reversed. Default is false. This is used for vertical text layout.
v-text-spacing-mode	Specifies the mode for letter spacing. Default is tightening. This property determines whether space is removed between each letter (tightening) or added between each letter (tracking). The amount of letter spacing change is defined by the v-text-spacing property. Allowed values are: • tightening • tracking
v-text-spacing	Specifies the amount of spacing for text in 100ths of single line spacing. Default is 100.

The line (§19.1.2.12), polyline (§19.1.2.15) and curve (§19.1.2.3) elements ignore the following properties:

- top
- left
- width
- height

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
The following properties are not inherited by an element that references a shapetype element (§19.1.2.20) via the id attribute:

flip
height
left
margin-left
margin-top
position
rotation
top
visibility
width
z-index

The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>target (Hyperlink Display Target)</td><td>
Specifies a frame or window that a URL is displayed in. Default is no value. Allowed values are:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><targetname></td><td>String containing the name of the frame or window in which to load the document.</td></tr>
<tr><td>_blank</td><td>Specifies that the linked document is loaded into a new blank window. This window is not named.</td></tr>
<tr><td>_media</td><td>Specifies that the linked document is loaded into the browser’s multimedia pane.</td></tr>
<tr><td>_parent</td><td>Specifies that the linked document is loaded into the immediate parent of the document containing the link.</td></tr>
<tr><td>_search</td><td>Specifies that the linked document is loaded into the browser's search pane.</td></tr>
<tr><td>_self</td><td>Specifies that the linked document is loaded into the window in which the link was clicked (the active window).</td></tr>
<tr><td>_top</td><td>Specifies that the linked document is loaded into the topmost window.</td></tr>
</table>
[Example:
<pre>
<v:shape …
  href="http://www.openxmlformats.org"
  target="_self" … >
</pre>
</td></tr>
</table>

Attributes	Description
	`</v:shape>` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

title (Shape Title)	Specifies the text displayed when the mouse pointer moves over the shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … title="tooltip" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

userdrawn (Exists In Master Slide)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the user has added the shape to a master slide. Default is `false`. Used by PresentationML.
	
	[*Example*:
	
	`<v:shape … o:userdrawn="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

userhidden (Hide Script Anchors)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a script anchor is hidden. Default is `false`. If `true`, script anchors stay hidden even if the shape is otherwise visible. A script anchor is the visual representation of a script that when displayed in an application.
	
	[*Example*:
	
	`<v:shape … o:userhidden="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

wrapcoords (Shape Bounding Polygon)	Specifies the bounding polygon that surrounds a shape. This is specified using a comma-delimited list of x and y coordinates: "x1,y1,x2,y2,x3,y3,…" This is used when text is

Attributes	Description

	tightly wrapped around a shape. Default is no value until the mso-wrap-mode style attribute is set to `tight` or `through`.
	
	[*Example*:
	
	`<v:shape …`
	`wrapcoords="0,0 0,200, 200,200, 200,0" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Image) is located in §A.6.1. *end note*]

19.1.2.11 imagedata (Image Data)

This element is used to draw an image that has been loaded from an external source. There is an implied rectangle that is the same size as the image. Any stroke or fill is applied to this implied rectangle. The stroke is drawn on top of the image. The fill is behind the image and therefore only visible through transparent areas of the image. Image transparency is either encoded in the file or defined via a color value using the chromakey attribute. Unlike the image element (§19.1.2.10), the imagedata element shall have a parent element.

[*Example*:

```
<v:shape style="position:relative;top:1;left:1;width:50;height:50"
  path="m 0,0 l 1000,0 1000,1000 0,1000 x e" fillcolor="blue">
  <v:imagedata src="myimage.gif"/>
</v:shape>
```

end example]

Attributes	Description

althref (Alternate Image Reference)
 	
Namespace: urn:schemas-microsoft-
	Defines an alternate reference for an image in Macintosh PICT format.
	
	[*Example*:
	
	`<v:imagedata … althref="myimage.pcz" … >`
	`</v:imagedata>`

Attributes	Description
com:office:office	*end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

bilevel (Image Bilevel Toggle)	Specifies that all colors in the picture shall be converted to either 0 or full intensity component values. This converts a color bitmap to 8 colors and a grayscale bitmap to black and white. Default is `false`.
	
	[*Example*:
	
	`<v:image … bilevel="true" …>`
	`</v:image>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

blacklevel (Image Brightness)	Specifies the image brightness. Default is 0.
	
	[*Example*:
	
	`<v:image … blacklevel="0.1" …>`
	`</v:image>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

chromakey (Image Transparency Color)	Specifies a color value that is transparent and show anything behind the shape. Default is no value.
	
	[*Example*:
	
	`<v:image … chromakey="white" …>`
	`</v:image>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

cropbottom (Image Bottom Crop)	Specifies the how much to crop the image from the bottom up as a fraction of picture size. Default is 0. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. For example, a value of "52429f" represents 52429/65536 or 0.8.
	
	[*Example*:
	
	`<v:image … cropbottom="10%" …>`
	`</v:image>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

cropleft (Image Left Crop)	Specifies how much to crop the image from the left in as a fraction of picture size. Default is 0. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. For example, a value of "52429f" represents 52429/65536 or 0.8.
	
	[*Example*:
	
	`<v:image … cropleft="10%" …>`
	`</v:image>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

cropright (Image Right Crop)	Specifies how much to crop the image from the right in as a fraction of picture size. Default is 0. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. [*Example*: For example, a value of "52429f" represents 52429/65536 or 0.8. *end example*]
	
	[*Example*:
	
	`<v:image … cropright="10%" …>`
	`</v:image>`

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema string datatype.

croptop (Image Top Crop)	Specifies how much to crop the image from the top down as a fraction of picture size. Default is 0. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. [*Example:* For example, a value of "52429f" represents 52429/65536 or 0.8. *end example*]
	
	[*Example*:
	
	`<v:image … croptop="10%" …>`
	`</v:image>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

detectmouseclick (Detect Mouse Click) Namespace: urn:schemas-microsoft-com:office:office	Specifies whether a mouse click is detected on the fill of a shape. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
embosscolor (Embossed Color)	Specifies the color to use to create an embossed effect in the image. Default is no value. This can be set to a percentage of the shadow color to create an embossed picture effect. The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).
gain (Image Intensity)	Specifies an adjustment for the intensity of all colors. Essentially sets how bright white is. Default is 1. [*Example*:

Attributes	Description

	`<v:image … gain="0.5" …>`
	`</v:image>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

gamma (Image Gamma Correction)	Specifies the gamma correction. Default is 1.
	
	Gamma correction is a factor by which the intended target display gamma differs from the sRGB profile. It can be used to correct for images not prepared for sRGB displays and to adjust overall image contrast. Decreasing it below 1 gives a higher contrast image.
	
	[*Example*:
	
	`<v:image … gamma="0.5" …>`
	`</v:image>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

grayscale (Image Grayscale Toggle)	Specifies to display the image in grayscale. Default is `false`.
	
	[*Example*:
	
	`<v:image … gamma="0.5" …>`
	`</v:image>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

href (Explicit	Specifies the relationship ID of the relationship to the hyperlink used for this VML object.

Attributes	Description
Relationship to Hyperlink Target) Namespace: …/officeDocument/2006/relationships	The specified relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/image or the document shall be considered non-conformant. [*Example*: The markup specifies the associated relationship part with relationship ID `rId10` contains the corresponding relationship information for the image data: `< … r:href="rId5" />` *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

href (Original Image Reference)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the URL to the original image file. Used only if the picture has been linked and embedded. Default is no value.
	
	[*Example*:
	
	`<v:fill … o:href="myimage.gif" … >`
	`</v:fill>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

id (Explicit Relationship to Image Data) Namespace: …/officeDocument/2006/relationships	Specifies the relationship ID of the relationship to the image used for this VML object. The specified relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/image or the document shall be considered non-conformant. [*Example*: The markup specifies the associated relationship part with relationship ID `rId10` contains the corresponding relationship information for the image data: `< … r:id="rId10" />` *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).
id (Unique Identifier)	Specifies a unique identifier that can be used to reference a VML object. Default is no value. [*Example*:

Attributes	Description

	`<v:shape … id="myShape" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

movie (Movie Reference)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies a pointer to a movie image. This is a data block that contains a pointer to a pointer to movie data.
	
	[*Example*:
	
	`<v:imagedata … o:movie="1434" …>`
	`</v:imagedata>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema float datatype.

oleid (Image Embedded Object ID) Namespace: urn:schemas-microsoft-com:office:office	Specifies the embedded object ID of an image. The possible values for this attribute are defined by the W3C XML Schema float datatype.
pict (Explicit Relationship to Alternate Image Data) Namespace: …/officeDocument/2006/relationships	Specifies the relationship ID of the relationship to an alternate format image used for this VML object. The specified relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/image or the document shall be considered non-conformant. If this attribute is specified, the application should attempt to display the image defined by the relationship. If the application cannot display the format of that image, the r:id attribute is used. [*Example*: The markup specifies the associated relationship part with relationship ID `rId7` contains the corresponding relationship information for the image data. The relationship part with relationship ID `rId10` is used if the application cannot display the image referenced by `rId7`.: `< … r:id="rId10" r:pict="rId7"/>` *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type

Attributes	Description
	(Part 1, §22.8.2.1).

recolortarget (Black Recoloring Color)	Specifies the color to which black should be recolored.
	
	[*Example*:
	
	`<v:imagedata r:id="rId4" recolortarget="red">`
	`</v:imagedata>`
	
	  `no recolor`
	  `recolortarget="red"`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

relid (Relationship to Part)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the relationship ID of the relationship to the image. The specified relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/image or the document shall be considered non-conformant.
	
	[*Example*: The markup specifies the associated relationship part with relationship ID `rId10` contains the corresponding relationship information:
	
	`<v:imagedata … o:relid="rId10" …>`
	`</v:imagedata>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

src (Image Source)	Specifies the URL of the image to use.
	
	[*Example*:
	
	`<v:image … src="myimage.gif" …>`
	`</v:image>`
	
	 

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema string datatype.

title (Image Data Title)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the title of an embedded image. This is typically set to the comment property of the image, which is often blank.
	
	[*Example*:
	
	`<v:fill … o:title="alt text" … >`
	`</v:fill>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ImageData) is located in §A.6.1. *end note*]

19.1.2.12 line (Line)

This element draws a straight line.

[*Example*:

```
<v:line from="10pt,10pt" to="75pt,35pt"
  strokecolor="blue" strokeweight="3pt">
</v:line>
```

end example]

Attributes	Description

allowincell (Allow in Table Cell)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can be placed in a table. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:allowincell="true" … >`
	`</v:shape>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

allowoverlap (Allow Shape Overlap)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can overlap another shape. Default is `true`. If `false`, the shape shifts left or right so as not to overlap another shape, similar to the behavior of the HTML float attribute.
	
	[*Example*:
	
	`<v:shape … o:allowoverlap="false" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

alt (Alternate Text)	Specifies alternative text describing the graphical object. This text should provide a brief description of the shape for use by accessibility tools. Default is no value.
	
	[*Example*: The alt text describes the basic shape:
	
	`<v:shape … fillcolor="red"`
	`  alt="Red rectangle">`
	`</v:shape>`
	
	The alt text describes the contents of a shape displaying an image:
	
	`<v:shape … alt="Picture of a sunset">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderbottomcolor (Bottom Border Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the bottom border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderbottomcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderleftcolor	Specifies the left border color of an inline shape. Default is no value.

Attributes	Description

(Border Left Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	[*Example*:
	
	`<v:shape … o:borderleftcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderrightcolor (Border Right Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the right border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderrightcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

bordertopcolor (Border Top Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the top border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:bordertopcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

bullet (Graphical Bullet)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the shape is a graphical bullet. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:bullet="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

button (Button Behavior Toggle)	Specifies whether a shape exhibits button press behavior on click. Default is `false`.

Attributes	Description

Namespace: urn:schemas-microsoft-com:office:office	[*Example*:
	
	`<v:shape … o:button="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

bwmode (Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies how a shape renders for black-and-white output devices. When a shape is printed on a black-and-white printer or displayed in a black-and-white view in an application, several options are possible. Default is `auto`, which uses o:bwnormal for normal black-and-white rendering and o:bwpure for pure black-and-white rendering.
	
	bwnormal and bwpure are subordinate to bwmode. If bwmode is "auto" then the value for bwnormal or bwpure is used depending on what the output format is. An application can define for itself what, if any, difference there is between normal B&W and pure B&W. [*Example*: Normal B&W might allow greyscale and pure B&W might not. *end example*]
	[*Example*: This shape renders in grayscale in a black-and-white environment:
	
	`<v:shape … o:bwmode="grayscale" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

bwnormal (Normal Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the black-and-white mode for normal black-and-white output devices. Default is `auto`.
	
	[*Example*: This shape renders in a pale grayscale in a normal black-and-white environment:
	
	`<v:shape …  o:bwmode="auto" o:bwnormal="lightgrayscale" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

bwpure (Pure Black-and-White Mode)	Specifies the black-and-white mode for pure black-and-white output devices. Default is `auto`. [*Example*: This shape renders in high contrast when in a pure black-and-white

Attributes	Description

Namespace: urn:schemas-microsoft-com:office:office	environment:
	
	`<v:shape … o:bwmode="auto" o:bwpure="highcontrast" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

chromakey (Image Transparency Color)	Specifies a color value that is transparent and show anything behind the shape. Default is no value.
	
	[*Example*:
	
	`<v:image … chromakey="white" …>`
	`</v:image>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

class (CSS Reference)	Specifies a reference to the definition of a CSS style. Default is no value.
	
	[*Example*: The snippets below are equivalent:
	
	`…`
	`.narrowstyle {width:50;height:100}`
	`…`
	`<v:shape … class="narrowstyle"`
	`  style="top:1;left:1">`
	`</v:shape>`
	
	`<v:shape … style="top:1;left:1;`
	`  width:50;height:100">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

clip (Clipping Toggle) Namespace: urn:schemas-	Specifies that the clipping region is active. This is used in conjunction with the clippath (§19.2.2.3) element to create a clipping region. [*Example*:

Attributes	Description

microsoft-com:office:office	`<v:shape … o:clip="true">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

cliptowrap (Clip to Wrapping Polygon)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the clipping region for the shape aligns with the wrapping polygon that tightly surrounds the entire shape (essentially, that the shape shall not be drawn beyond its wrapping polygon's extents – if it does, the shape shall be clipped). Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:cliptowrap="true">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

connectortype (Shape Connector Type)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of connector used for joining shapes. Default is `straight`.
	
	[*Example*:
	
	`<v:shape … o:connectortype="elbow" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ConnectorType simple type (§19.2.3.7).

coordorigin (Coordinate Space Origin)	Specifies the coordinate of the top left corner of the shape's coordinate space. This determines the position of the (0,0) coordinate space origin within the shape's bounding box. Default is "0,0", which places the (0,0) origin at the top left corner of the bounding box. This affects shape properties that specify coordinate positions, such as the path attribute. Thus a path can be defined against a generic (0,0) origin and the coordorigin value translates the entire path within the shape's bounding space. [*Example*: The horizontal and vertical coordinate space ranges from -100 to +100 because the coordinate space (coordsize) is 200 by 200 and the top left coordinate is (-100,-100). The (0,0) origin lies at the center of the shape's bounding box, as evidenced by the position of the shape's path within the coordinate space:

Attributes	Description
	<v:shape … coordsize="200,200" coordorigin="-100,-100" path="m 0,0 l 0,50, 50,50, 50,0 x e"> </v:shape> *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
coordsize (Coordinate Space Size)	Specifies the size of the shape's coordinate space in coordinate units. Default is "1000,1000". The physical size of a coordinate unit length is determined by both the size of the coordinate space (coordsize) and the size of the shape (style width and height). The coordsize attribute defines the number of horizontal and vertical subdivisions into which the shape's bounding box is divided. The combination of coordsize and style width/height effective scales the shape anisotropically. [*Example*: The path is 50 units wide and tall, which is 25% of the size of the coordinate space: <v:shape … coordsize="200,200" coordorigin="-100,-100" path="m 0,0 l 0,50, 50,50, 50,0 x e"> </v:shape> *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
dgmlayout (Diagram Node Layout Identifier) Namespace: urn:schemas-microsoft-	Specifies the kind of automatic layout to apply to the diagram node. This is only meaningful if the shape is a node in an organization chart, which is denoted by the orgchart value of the editas attribute of the group element. [*Example*: <v:shape … dgmlayout="1"> </v:shape>

Attributes	Description
com:office:office	*end example*] The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).

dgmlayoutmru (Diagram Node Recent Layout Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of automatic layout most recently used on the child elements of the diagram node. This is only meaningful if the shape is a node in anorganization chart, which is denoted by the `orgchart` value of the editas attribute of the group element.
	
	[*Example*:
	
	`<v:shape … dgmlayout="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).

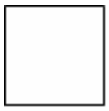
dgmnodekind (Diagram Node Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies an optional, application-defined parameter that is intended to be used by the application to tag different types of nodes in a diagram.
	
	[*Example*:
	
	`<v:shape … dgmnodekind="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

doubleclicknotify (Double-click Notification Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that an event message is sent when a shape is double-clicked. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:doubleclicknotify="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

fillcolor (Fill Color)	Specifies the color to use for the fill. Default is `white`. If the fill element (§19.1.2.5) is present, its color attribute takes precedence. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue

Attributes	Description

	values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*: This shape is red if its fill is visible:
	
	`<v:shape … fillcolor="red" … >`
	`</v:shape>`
	
	This is equivalent to:
	
	`<v:shape … fillcolor="#ff0000" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

filled (Shape Fill Toggle)	Specifies whether the closed path is filled. Default is `true`. This attribute is overridden by the fill on attribute.
	
	[*Example*:
	
	`<v:shape … filled="f"`
	`  fillcolor="red" …>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

forcedash (Force Dashed Outline)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a dashed outline is used to draw a shape when a shape has no line or fill. Default is `false`.
	
	Used by PresentationML placeholders to draw a dashed outline when there is no line and no fill for a shape.
	
	[*Example*:
	
	`<v:shape … o:forcedash="true" … >`
	`</v:shape>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

from (Line Start)	Specifies the starting point of the line in the coordinate space of the parent element. Default is "0,0". If the parent is not a VML element, the default unit is a pixel. Allowed units are in, cm, mm, pt, pc and px.
	
	[*Example*:
	
	`<v:line from="10pt,10pt" to="50pt,50pt">`
	`</v:line>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

hr (Horizontal Rule Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that a shape is a horizontal rule. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hr="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hralign (Horizontal Rule Alignment)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the alignment of a horizontal rule. Default is `left`.
	
	[*Example*:
	
	`<v:shape … o:hralign="center" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_HrAlign simple type (§19.2.3.16).

href (Hyperlink Target)	Specifies a hyperlink URL target for the shape. Default is no value. [*Example*:

Attributes	Description

	`<v:shape … href="http://www.openxmlformats.org" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

hrnoshade (Horizontal Rule 3D Shading Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the horizontal rule does not have 3-D shading. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hrnoshade="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hrpct (Horizontal Rule Length Percentage)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the length of a horizontal rule as a percentage of page width. Default is 0.
	
	[*Example*:
	
	`<v:shape … o:hrpct="85" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema float datatype.

hrstd (Horizontal Rule Standard Display Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape is displayed as a standard horizontal rule. Only applies if hr is `true`. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hrstd="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

id (Unique Identifier)	Specifies a unique identifier that can be used to reference a VML object. Default is no value.

Attributes	Description

	[*Example*:
	
	`<v:shape … id="myShape" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

insetmode (Text Inset Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the application calculates the internal text margin instead of using the inset attribute. Default is `custom`. This attribute is only meaningful for text boxes.
	
	[*Example*:
	
	`<v:shape … o:insetmode="auto" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_InsetMode simple type (§19.2.3.17).

insetpen (Inset Border From Path)	Specifies that the border shall be displayed inside of the path defining the shape, rather than along the path (the default border placement), or outside of the path as might be done with an image.
	
	[*Example*:
	
	`<v:shape … insetpen="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

ole (Embedded Object Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the shape is an embedded object. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:ole="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

Attributes	Description

oleicon (Embedded Object Icon Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether an embedded object is displayed as an icon. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oleicon="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

oned (Shape Handle Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the extra handles of a shape are hidden. If `true`, hides all shape handles except the top left and bottom right; that is, the same handles that are used for a straight line segment. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oned="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

opacity (Fill Color Opacity)	Specifies the opacity of the primary fill color. Default is 1.0. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. [*Example*: A value of "52429f" represents 52429/65536 or 0.8. *end example*]
	
	[*Example*: The red color is 25% opaque:
	
	`<v:fill type="gradient" color="red"`
	`color2="blue" opacity=".25">`
	`</v:fill>`
	
	  `opacity="1"`
	  `opacity=".25"`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

Attributes	Description

preferrelative (Relative Resize Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the original size of an object is saved after reformatting. Default is `false`. If `true`, the original size of the object is stored and all resizing is based on a percentage of that original size. Otherwise, each resizing resets the scale to 100%.
	
	[*Example*:
	
	`<v:shape … o:preferrelative="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

print (Print Toggle)	Specifies whether the shape is printed. Default is `true`.
	
	[*Example*:
	
	`<v:shape … print="false" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

regroupid (Regroup ID)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies a previous group for a shape. An ID number is used to identify groups of shapes that are no longer grouped. This allows shapes to be regrouped programmatically.
	
	[*Example*: The shape was part of a group identified by the ID 040754:
	
	`<v:shape … o:regroupid="040754" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

spid (Optional String) Namespace: urn:schemas-microsoft-com:office:office	Specifies an optional string that an application can use to Identify the particular shape. Default is no value. The possible values for this attribute are defined by the W3C XML Schema string datatype.
spt (Optional Number)	Specifies an optional number that an application can use to associate the particular shape with a defined shape type. Default is 0.

Attributes	Description
Namespace: urn:schemas-microsoft-com:office:office	The possible values for this attribute are defined by the W3C XML Schema float datatype.

strokecolor (Shape Stroke Color)	Specifies the primary color of the brush to use to stroke the path of the shape. Default is `black`. The color attribute of the stroke element (§19.1.2.21) overrides this. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*:
	
	`<v:shape … strokecolor="red" …>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

stroked (Shape Stroke Toggle)	Specifies whether the path defining the shape is stroked with a solid line. The stroke element (§19.1.2.21) defines other strokes. The on attribute of the stroke element overrides this attribute. Default is `true`.
	
	[*Example*:
	
	`<v:shape … fillcolor="red"`
	`  stroked="false" strokecolor="blue"…>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

strokeweight (Shape Stroke Weight)	Specifies the width of the brush to use to stroke the path. Default is 1 point. If a number is given without units, the emu is used. The weight attribute of the stroke element (§19.1.2.21) overrides this attribute.

Attributes	Description

	[*Example*:
	
	`<v:shape … strokeweight="3pt" … >`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

style (Shape Styling Properties)	Specifies the CSS2 styling properties of the shape. The CSS2 (Cascading Style Sheets, Level 2) specification, a Recommendation of the World Wide Web Consortium, is available here: http://www.w3.org/TR/REC-CSS2.
	
	This attribute uses a semi-colon delimited list of "name:value" pairs, the syntax defined by section 4.1.8 of the CSS2 specification for grouped declarations (without the surrounding braces). The following tables define the allowed properties and the VML treatment of each.
	
	[*Example*:
	`<v:shape … style='position:absolute;width:100pt;height:50pt' … >`
	`</v:shape>`
	end example]
	
	Property / **Description**
	`flip` — Specifies that the orientation of a shape is flipped. Default is no value. Allowed values are:
	• x - Flip along the y-axis, reversing the x-coordinates.
	• y - Flip along the x-axis, reversing the y-coordinates.
	• xy - Flip along both the y- and x-axis.
	• yx - Flip along both the x- and y-axis.
	
	`height` — Specifies the height of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are:
	• `auto` - Default position of an element in the flow of the page.
	• <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or

Attributes	Description	
		ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`left`	Specifies the position of the left of the containing block of the shape relative to the element left of it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-bottom`	Specifies the position of the bottom of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`margin-left`	Specifies the position of the left of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-right`	Specifies the position of the right of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are:

Attributes	Description	
		• auto - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	margin-top	Specifies the position of the top of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • auto - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	mso-position-horizontal	Specifies the horizontal positioning data for objects in WordprocessingML documents. Default is absolute. Allowed values are: • absolute • left • center • right • inside • outside
	mso-position-horizontal-relative	Specifies relative horizontal position data for objects in WordprocessingML documents. This modifies the mso-position-horizontal property. Default is text. Allowed values are: • margin • page • text • char
	mso-position-vertical	Specifies the vertical positioning data for objects in WordprocessingML documents. Default is absolute. Allowed values are:

Attributes	Description
	• absolute • top • center • bottom • inside • outside
mso-position-vertical-relative	Specifies relative vertical position data for objects in WordprocessingML documents. This modifies the mso-position-vertical property. Default is text. Allowed values are: • margin • page • text • line
mso-wrap-distance-bottom	Specifies the distance from the bottom of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
mso-wrap-distance-left	Specifies the distance from the left side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
mso-wrap-distance-right	Specifies the distance from the right side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
mso-wrap-distance-top	Specifies the distance from the top of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
mso-wrap-edited	Specifies whether the wrap coordinates were customized by the user. If the wrap coordinates are generated by an editor, this property is true; otherwise they were customized by a user. Default is false.
mso-wrap-style	Specifies the wrapping mode for text in shapes in WordprocessingML documents. Default is square. Allowed

Attributes	Description	
		values are: • `square` - Wraps text inside the shape in a square. • `none` - Text does not wrap.
	`position`	Specifies the kind of positioning used to place an element. Default is `static`. When the element is contained inside a group, this property shall be `absolute`. Allowed values are: • `static` - The element is positioned according to the normal flow of the page. The `top` and `left` properties are ignored. If the object is anchored inline, this value is used. • `absolute` - The element is positioned relative to the parent, using the `top` and `left` properties. • `relative` - The element is positioned according to the normal flow of the page, but the `top` and `left` properties are used. The overlap of overlapping elements is governed by the `z-index` property.
	`rotation`	Specifies the angle that a shape is rotated, in degrees. Default is 0. Positive angles are clockwise.
	`top`	Specifies the position of the top of the containing block of the shape relative to the element above it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`visibility`	Specifies whether a shape is displayed. Only `inherit` and `hidden` are used; any other values are mapped to `inherit`. Default is `inherit`. Allowed values are: • `hidden` - The shape is not visible, but is still part of the flow of the objects in the browser. Mouse events are not processed. • `inherit` - The visibility state is inherited from the parent of the shape.

Attributes	Description
	`width` — Specifies the width of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`z-index` — Specifies the display order of overlapping shapes. Default is 0. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Uses the order that the shapes appear in the page, bottom to top. • <order>- A number that represents the stacking precedence. Shapes with higher numbers are placed on top of those with lower numbers. Negative numbers are allowed.

The following properties are only used by the textbox element (§19.1.2.22):

Property	Description
`direction`	Specifies the direction of the text in the textbox. Default is `ltr`. This property is superceded by the `mso-direction-alt` property if that is specified. Allowed values are: • `ltr` - Text is displayed left-to-right. • `rtl` - Text is displayed right-to-left.
`layout-flow`	Determines the flow of the text layout in a textbox. Default is `horizontal`. Allowed values are: • `horizontal` - Text is displayed horizontally. • `vertical` - Text is displayed vertically. • `vertical-ideographic` - Ideographic text is displayed vertically. • `horizontal-ideographic` - Ideographic text is displayed horizontally.
`mso-direction-`	Specifies an alternate direction for text in textboxes. Overrides

Attributes	Description
	alt — the direction property. The only allowed value is context. mso-fit-shape-to-text — Specifies whether the shape stretches to fit the text in the textbox. Default is false. mso-fit-text-to-shape — Specifies whether the text stretches to fit the textbox. Default is false. mso-layout-flow-alt — Specifies the alternate layout flow for text in textboxes. This property is used instead of layout-flow when the layout flow is from bottom to top for non-ideographic languages. Its only value is bottom-to-top. mso-next-textbox — Specifies the ID of the next textbox in a series. Used to keep track of a set of linked textboxes. Default is no value. mso-rotate — Specifies a specific rotation value for text in a textbox. Default is 0. Allowed values are: • 0 • 90 • 180 • -90 mso-text-scale — Specifies the scaling factor for fitting text to shapes. Default is 0. This property is only used if mso-fit-text-to-shape is true. v-text-anchor — Specifies the vertical anchoring of text in a textbox. Default is top. The alignment of a text anchor only becomes evident if mso-fit-text-to-shape is false. This property is different from the vertical-align CSS property, which is used for ideographic languages. Allowed values are: • top • middle • bottom • top-center • middle-center • bottom-center • top-baseline • bottom-baseline • top-center-baseline • bottom-center-baseline The following properties are only used by the textpath element (§19.1.2.23):

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Property</th><th>Description</th></tr>
<tr><td>font</td><td>Specifies a compound value of font settings. Default is no value. The values are the same as those of the CSS font property. The order of definitions in the string is: font-style, font-variant, font-weight, font-size, line-height, font-family.</td></tr>
<tr><td>font-family</td><td>Specifies the family of the font. Default is no value. The values are the same as those of the CSS font-family property.</td></tr>
<tr><td>font-size</td><td>Specifies the size of the font. Default is no value. The font size is defined in points. The values are the same as those of the CSS font-size property.</td></tr>
<tr><td>font-style</td><td>Specifies the amount of slant for a font. Default is normal. The values are the same as those of the CSS font-style property. Allowed values are:

• normal
• italic
• oblique - Treated the same as italic.</td></tr>
<tr><td>font-variant</td><td>Specifies the variant style of a font. Default is normal. The values are the same as those of the CSS font-variant property. Allowed values are:

• normal
• small-caps</td></tr>
<tr><td>font-weight</td><td>Specifies the thickness of the letters of the font. Default is normal. The values are the same as those of the CSS font-weight property. Allowed values are:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>normal
lighter
100
200
300
400</td><td>Treated as non-bold.</td></tr>
<tr><td>bold
bolder</td><td>Treated as bold.</td></tr>
</table></td></tr>
</table>
</td></tr>
</table>

Attributes	Description
	500 600 700 800 900
	`mso-text-shadow` — Specifies whether a shadow is applied to the text on a text path. Default is `false`.
	`text-decoration` — Specifies the style of text decoration. Default is `none`. The values are the same as those of the CSS text-decoration property. Allowed values are: • `none` • `underline` • `overline` • `line-through` • `blink`
	`v-rotate-letters` — Specifies whether the letters of the text are rotated counterclockwise by 90 degrees. Default is `false`.
	`v-same-letter-heights` — Specifies whether all letters are the same height regardless of initial case. If `true`, the lowercase letters are stretched to the height of the uppercase letters. Default is `false`.
	`v-text-align` — Specifies the alignment of text. Default is `left`. Allowed values are: • `left` • `right` • `center` • `justify` • `letter-justify` - Distributes the extra space between the letters. • `stretch-justify` - Stretches the letters to fill in the space.
	`v-text-kern` — Specifies whether kerning is turned on. Default is `false`.
	`v-text-reverse` — Specifies whether the layout order of rows is reversed. Default is `false`. This is used for vertical text layout.
	`v-text-` — Specifies the mode for letter spacing. Default is `tightening`. This property determines whether space is removed between

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>spacing-mode</td><td>each letter (tightening) or added between each letter (tracking). The amount of letter spacing change is defined by the v-text-spacing property. Allowed values are:
• tightening
• tracking</td></tr>
<tr><td>v-text-spacing</td><td>Specifies the amount of spacing for text in 100ths of single line spacing. Default is 100.</td></tr>
</table>

The line (§19.1.2.12), polyline (§19.1.2.15) and curve (§19.1.2.3) elements ignore the following properties:
• top
• left
• width
• height

The following properties are not inherited by an element that references a shapetype element (§19.1.2.20) via the id attribute:
• flip
• height
• left
• margin-left
• margin-top
• position
• rotation
• top
• visibility
• width
• z-index

The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>target (Hyperlink Display Target)</td><td>Specifies a frame or window that a URL is displayed in. Default is no value. Allowed values are:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><targetname></td><td>String containing the name of the frame or window in which to load the document.</td></tr>
<tr><td>_blank</td><td>Specifies that the linked document is loaded into a new blank window. This window is not named.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>_media</td><td>Specifies that the linked document is loaded into the browser’s multimedia pane.</td></tr>
<tr><td>_parent</td><td>Specifies that the linked document is loaded into the immediate parent of the document containing the link.</td></tr>
<tr><td>_search</td><td>Specifies that the linked document is loaded into the browser's search pane.</td></tr>
<tr><td>_self</td><td>Specifies that the linked document is loaded into the window in which the link was clicked (the active window).</td></tr>
<tr><td>_top</td><td>Specifies that the linked document is loaded into the topmost window.</td></tr>
</table>
[Example:
<pre>
<v:shape …
  href="http://www.openxmlformats.org"
  target="_self" … >
</v:shape>
</pre>
end example]
The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>title (Shape Title)</td><td>
Specifies the text displayed when the mouse pointer moves over the shape. Default is no value.
[Example:
<pre>
<v:shape … title="tooltip" … >
</v:shape>
</pre>
end example]
The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>to (Line End Point)</td><td>
Specifies the ending point of the line in the coordinate space of the parent element. Default is "10,10". If the parent is not a VML element, the default unit is a pixel. Allowed units are in, cm, mm, pt, pc and px.
[Example:
<pre>
<v:line from="10pt,10pt" to="50pt,50pt">
</v:line>
</pre>
</td></tr>
</table>

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema string datatype.

userdrawn (Exists In Master Slide)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the user has added the shape to a master slide. Default is `false`. Used by PresentationML.
	
	[*Example*:
	
	`<v:shape … o:userdrawn="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

userhidden (Hide Script Anchors)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a script anchor is hidden. Default is `false`. If `true`, script anchors stay hidden even if the shape is otherwise visible. A script anchor is the visual representation of a script that when displayed in an application.
	
	[*Example*:
	
	`<v:shape … o:userhidden="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

wrapcoords (Shape Bounding Polygon)	Specifies the bounding polygon that surrounds a shape. This is specified using a comma-delimited list of x and y coordinates: "x1,y1,x2,y2,x3,y3,…" This is used when text is tightly wrapped around a shape. Default is no value until the mso-wrap-mode style attribute is set to `tight` or `through`.
	
	[*Example*:
	
	`<v:shape …`
	`  wrapcoords="0,0 0,200, 200,200, 200,0" … >`
	`</v:shape>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Line) is located in §A.6.1. *end note*]

19.1.2.13 oval (Oval)

This element draws an oval sized according to the CSS2 style content width and height.

[*Example*:

```
<v:oval fillcolor="blue"
  style="position:relative;top:1;left:1;width:150;height:50">
</v:oval>
```

end example]

Attributes	Description

allowincell (Allow in Table Cell)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can be placed in a table. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:allowincell="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

allowoverlap (Allow Shape Overlap)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can overlap another shape. Default is `true`. If `false`, the shape shifts left or right so as not to overlap another shape, similar to the behavior of the HTML float attribute.
	
	[*Example*:
	
	`<v:shape … o:allowoverlap="false" … >`
	`</v:shape>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

alt (Alternate Text)	Specifies alternative text describing the graphical object. This text should provide a brief description of the shape for use by accessibility tools. Default is no value.
	
	[*Example*: The alt text describes the basic shape:
	
	`<v:shape … fillcolor="red"`
	`  alt="Red rectangle">`
	`</v:shape>`
	
	The alt text describes the contents of a shape displaying an image:
	
	`<v:shape … alt="Picture of a sunset">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderbottomcolor (Bottom Border Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the bottom border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderbottomcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderleftcolor (Border Left Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the left border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderleftcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderrightcolor (Border Right Color)	Specifies the right border color of an inline shape. Default is no value. [*Example*:

Attributes	Description

Namespace: urn:schemas-microsoft-com:office:office	`<v:shape … o:borderrightcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

bordertopcolor (Border Top Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the top border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:bordertopcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

bullet (Graphical Bullet)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the shape is a graphical bullet. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:bullet="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

button (Button Behavior Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape exhibits button press behavior on click. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:button="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

bwmode (Black-and-White Mode) Namespace:	Specifies how a shape renders for black-and-white output devices. When a shape is printed on a black-and-white printer or displayed in a black-and-white view in an application, several options are possible. Default is `auto`, which uses o:bwnormal for normal black-and-white rendering and o:bwpure for pure black-and-white rendering.

Attributes	Description

urn:schemas-microsoft-com:office:office	bwnormal and bwpure are subordinate to bwmode. If bwmode is "auto" then the value for bwnormal or bwpure is used depending on what the output format is. An application can define for itself what, if any, difference there is between normal B&W and pure B&W. [*Example*: Normal B&W might allow greyscale and pure B&W might not. *end example*]
	[*Example*: This shape renders in grayscale in a black-and-white environment:
	
	`<v:shape … o:bwmode="grayscale" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

bwnormal (Normal Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the black-and-white mode for normal black-and-white output devices. Default is `auto`.
	
	[*Example*: This shape renders in a pale grayscale in a normal black-and-white environment:
	
	`<v:shape … o:bwmode="auto" o:bwnormal="lightgrayscale" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

bwpure (Pure Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the black-and-white mode for pure black-and-white output devices. Default is `auto`.
	
	[*Example*: This shape renders in high contrast when in a pure black-and-white environment:
	
	`<v:shape … o:bwmode="auto" o:bwpure="highcontrast" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

chromakey (Image Transparency Color)	Specifies a color value that is transparent and show anything behind the shape. Default is no value. [*Example*:

Attributes	Description

	`<v:image … chromakey="white" …>`
	`</v:image>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

class (CSS Reference)	Specifies a reference to the definition of a CSS style. Default is no value.
	
	[*Example*: The snippets below are equivalent:
	
	`…`
	`.narrowstyle {width:50;height:100}`
	`…`
	`<v:shape … class="narrowstyle"`
	`  style="top:1;left:1">`
	`</v:shape>`
	
	`<v:shape … style="top:1;left:1;`
	`  width:50;height:100">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

clip (Clipping Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the clipping region is active. This is used in conjunction with the clippath (§19.2.2.3) element to create a clipping region.
	
	[*Example*:
	
	`<v:shape … o:clip="true">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

cliptowrap (Clip to Wrapping Polygon) Namespace: urn:schemas-microsoft-	Specifies that the clipping region for the shape aligns with the wrapping polygon that tightly surrounds the entire shape (essentially, that the shape shall not be drawn beyond its wrapping polygon's extents – if it does, the shape shall be clipped). Default is `false`. [*Example*:

Attributes	Description

com:office:office	
	`<v:shape … o:cliptowrap="true">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

connectortype (Shape Connector Type)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of connector used for joining shapes. Default is `straight`.
	
	[*Example*:
	
	`<v:shape … o:connectortype="elbow" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ConnectorType simple type (§19.2.3.7).

coordorigin (Coordinate Space Origin)	Specifies the coordinate of the top left corner of the shape's coordinate space. This determines the position of the (0,0) coordinate space origin within the shape's bounding box. Default is "0,0", which places the (0,0) origin at the top left corner of the bounding box.
	
	This affects shape properties that specify coordinate positions, such as the path attribute. Thus a path can be defined against a generic (0,0) origin and the coordorigin value translates the entire path within the shape's bounding space.
	
	[*Example*: The horizontal and vertical coordinate space ranges from -100 to +100 because the coordinate space (coordsize) is 200 by 200 and the top left coordinate is (-100,-100). The (0,0) origin lies at the center of the shape's bounding box, as evidenced by the position of the shape's path within the coordinate space:
	
	`<v:shape … coordsize="200,200"`
	`  coordorigin="-100,-100"`
	`  path="m 0,0 l 0,50, 50,50, 50,0 x e">`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

Attributes	Description

coordsize (Coordinate Space Size)	Specifies the size of the shape's coordinate space in coordinate units. Default is "1000,1000".
	
	The physical size of a coordinate unit length is determined by both the size of the coordinate space (coordsize) and the size of the shape (style width and height). The coordsize attribute defines the number of horizontal and vertical subdivisions into which the shape's bounding box is divided. The combination of coordsize and style width/height effective scales the shape anisotropically.
	
	[*Example*: The path is 50 units wide and tall, which is 25% of the size of the coordinate space:
	
	`<v:shape … coordsize="200,200"`
	`  coordorigin="-100,-100"`
	`  path="m 0,0 l 0,50, 50,50, 50,0 x e">`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

dgmlayout (Diagram Node Layout Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of automatic layout to apply to the diagram node. This is only meaningful if the shape is a node in an organization chart, which is denoted by the `orgchart` value of the editas attribute of the group element.
	[*Example*:
	
	`<v:shape … dgmlayout="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).

dgmlayoutmru (Diagram Node Recent Layout Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of automatic layout most recently used on the child elements of the diagram node. This is only meaningful if the shape is a node in anorganization chart, which is denoted by the `orgchart` value of the editas attribute of the group element.
	
	[*Example*:
	
	`<v:shape … dgmlayout="1">`
	`</v:shape>`

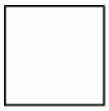
Attributes	Description
	end example] The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).

dgmnodekind (Diagram Node Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies an optional, application-defined parameter that is intended to be used by the application to tag different types of nodes in a diagram.
	
	[*Example*:
	
	`<v:shape … dgmnodekind="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

doubleclicknotify (Double-click Notification Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that an event message is sent when a shape is double-clicked. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:doubleclicknotify="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

fillcolor (Fill Color)	Specifies the color to use for the fill. Default is `white`. If the fill element (§19.1.2.5) is present, its color attribute takes precedence. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*: This shape is red if its fill is visible:
	
	`<v:shape … fillcolor="red" … >`
	`</v:shape>`
	
	This is equivalent to:
	
	`<v:shape … fillcolor="#ff0000" … >`
	`</v:shape>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

filled (Shape Fill Toggle)	Specifies whether the closed path is filled. Default is `true`. This attribute is overridden by the fill on attribute.
	
	[*Example*:
	
	`<v:shape … filled="f"`
	`  fillcolor="red" …>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

forcedash (Force Dashed Outline)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a dashed outline is used to draw a shape when a shape has no line or fill. Default is `false`.
	
	Used by PresentationML placeholders to draw a dashed outline when there is no line and no fill for a shape.
	
	[*Example*:
	
	`<v:shape … o:forcedash="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hr (Horizontal Rule Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that a shape is a horizontal rule. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hr="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type

Attributes	Description
	(§20.1.2.5).

hralign (Horizontal Rule Alignment)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the alignment of a horizontal rule. Default is `left`.
	
	[*Example*:
	
	`<v:shape … o:hralign="center" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_HrAlign simple type (§19.2.3.16).

href (Hyperlink Target)	Specifies a hyperlink URL target for the shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … href="http://www.openxmlformats.org" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

hrnoshade (Horizontal Rule 3D Shading Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the horizontal rule does not have 3-D shading. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hrnoshade="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hrpct (Horizontal Rule Length Percentage)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the length of a horizontal rule as a percentage of page width. Default is 0.
	
	[*Example*:
	
	`<v:shape … o:hrpct="85" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema float datatype.

hrstd (Horizontal	Specifies whether a shape is displayed as a standard horizontal rule. Only applies if hr is

Attributes	Description

Rule Standard Display Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	`true`. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hrstd="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

id (Unique Identifier)	Specifies a unique identifier that can be used to reference a VML object.
	
	Default is no value.
	
	[*Example*:
	
	`<v:shape … id="myShape" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

insetmode (Text Inset Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the application calculates the internal text margin instead of using the inset attribute. Default is `custom`. This attribute is only meaningful for text boxes.
	
	[*Example*:
	
	`<v:shape … o:insetmode="auto" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_InsetMode simple type (§19.2.3.17).

insetpen (Inset Border From Path)	Specifies that the border shall be displayed inside of the path defining the shape, rather than along the path (the default border placement), or outside of the path as might be done with an image.
	
	[*Example*:
	
	`<v:shape … insetpen="true" … >`
	`</v:shape>`

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

ole (Embedded Object Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the shape is an embedded object. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:ole="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

oleicon (Embedded Object Icon Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether an embedded object is displayed as an icon. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oleicon="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

oned (Shape Handle Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the extra handles of a shape are hidden. If `true`, hides all shape handles except the top left and bottom right; that is, the same handles that are used for a straight line segment. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oned="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

opacity (Fill Color Opacity)	Specifies the opacity of the primary fill color. Default is 1.0. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. [*Example*: A value of "52429f" represents 52429/65536 or 0.8. *end example*] [*Example*: The red color is 25% opaque:

Attributes	Description

	`<v:fill type="gradient" color="red"`
	`  color2="blue" opacity=".25">`
	`</v:fill>`
	
	  `opacity="1"`
	  `opacity=".25"`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

preferrelative (Relative Resize Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the original size of an object is saved after reformatting. Default is `false`. If `true`, the original size of the object is stored and all resizing is based on a percentage of that original size. Otherwise, each resizing resets the scale to 100%.
	
	[*Example*:
	
	`<v:shape … o:preferrelative="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

print (Print Toggle)	Specifies whether the shape is printed. Default is `true`.
	
	[*Example*:
	
	`<v:shape … print="false" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

regroupid (Regroup ID) Namespace: urn:schemas-microsoft-	Specifies a previous group for a shape. An ID number is used to identify groups of shapes that are no longer grouped. This allows shapes to be regrouped programmatically. [*Example*: The shape was part of a group identified by the ID 040754: `<v:shape … o:regroupid="040754" … >`

Attributes	Description
com:office:office	`</v:shape>` *end example*] The possible values for this attribute are defined by the W3C XML Schema integer datatype.
spid (Optional String) Namespace: urn:schemas-microsoft-com:office:office	Specifies an optional string that an application can use to Identify the particular shape. Default is no value. The possible values for this attribute are defined by the W3C XML Schema string datatype.
spt (Optional Number) Namespace: urn:schemas-microsoft-com:office:office	Specifies an optional number that an application can use to associate the particular shape with a defined shape type. Default is 0. The possible values for this attribute are defined by the W3C XML Schema float datatype.

strokecolor (Shape Stroke Color)	Specifies the primary color of the brush to use to stroke the path of the shape. Default is `black`. The color attribute of the stroke element (§19.1.2.21) overrides this. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*:
	
	`<v:shape … strokecolor="red" …>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

stroked (Shape Stroke Toggle)	Specifies whether the path defining the shape is stroked with a solid line. The stroke element (§19.1.2.21) defines other strokes. The on attribute of the stroke element overrides this attribute. Default is `true`. [*Example*:

Attributes	Description

	`<v:shape … fillcolor="red"`
	`  stroked="false" strokecolor="blue"…>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

strokeweight (Shape Stroke Weight)	Specifies the width of the brush to use to stroke the path. Default is 1 point. If a number is given without units, the emu is used. The weight attribute of the stroke element (§19.1.2.21) overrides this attribute.
	
	[*Example*:
	
	`<v:shape … strokeweight="3pt" … >`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

style (Shape Styling Properties)	Specifies the CSS2 styling properties of the shape. The CSS2 (Cascading Style Sheets, Level 2) specification, a Recommendation of the World Wide Web Consortium, is available here: http://www.w3.org/TR/REC-CSS2.
	
	This attribute uses a semi-colon delimited list of “name:value” pairs, the syntax defined by section 4.1.8 of the CSS2 specification for grouped declarations (without the surrounding braces). The following tables define the allowed properties and the VML treatment of each.
	
	[*Example*:
	`<v:shape … style=’position:absolute;width:100pt;height:50pt’ …`
	`>`
	`</v:shape>`
	end example]
	
	Property | **Description**

Attributes	Description	
	`flip`	Specifies that the orientation of a shape is flipped. Default is no value. Allowed values are: • x - Flip along the y-axis, reversing the x-coordinates. • y - Flip along the x-axis, reversing the y-coordinates. • xy - Flip along both the y- and x-axis. • yx - Flip along both the x- and y-axis.
	`height`	Specifies the height of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`left`	Specifies the position of the left of the containing block of the shape relative to the element left of it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-bottom`	Specifies the position of the bottom of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.

Attributes	Description	
	`margin-left`	Specifies the position of the left of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-right`	Specifies the position of the right of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-top`	Specifies the position of the top of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`mso-position-horizontal`	Specifies the horizontal positioning data for objects in WordprocessingML documents. Default is `absolute`. Allowed values are: • `absolute` • `left` • `center` • `right`

Attributes	Description
	• inside • outside
mso-position-horizontal-relative	Specifies relative horizontal position data for objects in WordprocessingML documents. This modifies the mso-position-horizontal property. Default is text. Allowed values are: • margin • page • text • char
mso-position-vertical	Specifies the vertical positioning data for objects in WordprocessingML documents. Default is absolute. Allowed values are: • absolute • top • center • bottom • inside • outside
mso-position-vertical-relative	Specifies relative vertical position data for objects in WordprocessingML documents. This modifies the mso-position-vertical property. Default is text. Allowed values are: • margin • page • text • line
mso-wrap-distance-bottom	Specifies the distance from the bottom of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
mso-wrap-distance-left	Specifies the distance from the left side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
mso-wrap-	Specifies the distance from the right side of the shape to the text

Attributes	Description
	`distance-right` — that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	`mso-wrap-distance-top` — Specifies the distance from the top of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	`mso-wrap-edited` — Specifies whether the wrap coordinates were customized by the user. If the wrap coordinates are generated by an editor, this property is `true`; otherwise they were customized by a user. Default is `false`.
	`mso-wrap-style` — Specifies the wrapping mode for text in shapes in WordprocessingML documents. Default is `square`. Allowed values are: • `square` - Wraps text inside the shape in a square. • `none` - Text does not wrap.
	`position` — Specifies the kind of positioning used to place an element. Default is `static`. When the element is contained inside a group, this property shall be `absolute`. Allowed values are: • `static` - The element is positioned according to the normal flow of the page. The `top` and `left` properties are ignored. If the object is anchored inline, this value is used. • `absolute` - The element is positioned relative to the parent, using the `top` and `left` properties. • `relative` - The element is positioned according to the normal flow of the page, but the `top` and `left` properties are used. The overlap of overlapping elements is governed by the `z-index` property.
	`rotation` — Specifies the angle that a shape is rotated, in degrees. Default is 0. Positive angles are clockwise.
	`top` — Specifies the position of the top of the containing block of the shape relative to the element above it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page.

Attributes	Description
	<table><tr><td></td><td>• <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.</td></tr><tr><td>`visibility`</td><td>Specifies whether a shape is displayed. Only `inherit` and `hidden` are used; any other values are mapped to `inherit`. Default is `inherit`. Allowed values are: • `hidden` - The shape is not visible, but is still part of the flow of the objects in the browser. Mouse events are not processed. • `inherit` - The visibility state is inherited from the parent of the shape.</td></tr><tr><td>`width`</td><td>Specifies the width of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.</td></tr><tr><td>`z-index`</td><td>Specifies the display order of overlapping shapes. Default is 0. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Uses the order that the shapes appear in the page, bottom to top. • <order>- A number that represents the stacking precedence. Shapes with higher numbers are placed on top of those with lower numbers. Negative numbers are allowed.</td></tr></table> The following properties are only used by the textbox element (§19.1.2.22): <table><tr><th>Property</th><th>Description</th></tr><tr><td>`direction`</td><td>Specifies the direction of the text in the textbox. Default is `ltr`. This property is superceded by the `mso-direction-alt` property if that is specified. Allowed values are:</td></tr></table>

Attributes	Description	
		• `ltr` - Text is displayed left-to-right. • `rtl` - Text is displayed right-to-left.
	`layout-flow`	Determines the flow of the text layout in a textbox. Default is `horizontal`. Allowed values are: • `horizontal` - Text is displayed horizontally. • `vertical` - Text is displayed vertically. • `vertical-ideographic` - Ideographic text is displayed vertically. • `horizontal-ideographic` - Ideographic text is displayed horizontally.
	`mso-direction-alt`	Specifies an alternate direction for text in textboxes. Overrides the `direction` property. The only allowed value is `context`.
	`mso-fit-shape-to-text`	Specifies whether the shape stretches to fit the text in the textbox. Default is `false`.
	`mso-fit-text-to-shape`	Specifies whether the text stretches to fit the textbox. Default is `false`.
	`mso-layout-flow-alt`	Specifies the alternate layout flow for text in textboxes. This property is used instead of `layout-flow` when the layout flow is from bottom to top for non-ideographic languages. Its only value is `bottom-to-top`.
	`mso-next-textbox`	Specifies the ID of the next textbox in a series. Used to keep track of a set of linked textboxes. Default is no value.
	`mso-rotate`	Specifies a specific rotation value for text in a textbox. Default is 0. Allowed values are: • 0 • 90 • 180 • -90
	`mso-text-scale`	Specifies the scaling factor for fitting text to shapes. Default is 0. This property is only used if `mso-fit-text-to-shape` is `true`.
	`v-text-anchor`	Specifies the vertical anchoring of text in a textbox. Default is `top`. The alignment of a text anchor only becomes evident if `mso-fit-text-to-shape` is `false`. This property is different from the `vertical-align` CSS property, which is used for ideographic languages. Allowed values are:

Attributes	Description
	<table><tr><td></td><td>• top • middle • bottom • top-center • middle-center • bottom-center • top-baseline • bottom-baseline • top-center-baseline • bottom-center-baseline</td></tr></table> The following properties are only used by the textpath element (§19.1.2.23): <table><tr><th>Property</th><th>Description</th></tr><tr><td>font</td><td>Specifies a compound value of font settings. Default is no value. The values are the same as those of the CSS font property. The order of definitions in the string is: font-style, font-variant, font-weight, font-size, line-height, font-family.</td></tr><tr><td>font-family</td><td>Specifies the family of the font. Default is no value. The values are the same as those of the CSS font-family property.</td></tr><tr><td>font-size</td><td>Specifies the size of the font. Default is no value. The font size is defined in points. The values are the same as those of the CSS font-size property.</td></tr><tr><td>font-style</td><td>Specifies the amount of slant for a font. Default is normal. The values are the same as those of the CSS font-style property. Allowed values are: • normal • italic • oblique - Treated the same as italic.</td></tr><tr><td>font-variant</td><td>Specifies the variant style of a font. Default is normal. The values are the same as those of the CSS font-variant property. Allowed values are: • normal • small-caps</td></tr><tr><td>font-weight</td><td>Specifies the thickness of the letters of the font. Default is normal. The values are the same as those of the CSS font-weight property. Allowed values are: <table><tr><th>Value</th><th>Description</th></tr></table></td></tr></table>

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>
<table>
<tr><td>normal
lighter
100
200
300
400</td><td>Treated as non-bold.</td></tr>
<tr><td>bold
bolder
500
600
700
800
900</td><td>Treated as bold.</td></tr>
</table>
</td></tr>
<tr><td>mso-text-shadow</td><td>Specifies whether a shadow is applied to the text on a text path. Default is false.</td></tr>
<tr><td>text-decoration</td><td>Specifies the style of text decoration. Default is none. The values are the same as those of the CSS text-decoration property. Allowed values are:
• none
• underline
• overline
• line-through
• blink</td></tr>
<tr><td>v-rotate-letters</td><td>Specifies whether the letters of the text are rotated counterclockwise by 90 degrees. Default is false.</td></tr>
<tr><td>v-same-letter-heights</td><td>Specifies whether all letters are the same height regardless of initial case. If true, the lowercase letters are stretched to the height of the uppercase letters. Default is false.</td></tr>
<tr><td>v-text-align</td><td>Specifies the alignment of text. Default is left. Allowed values are:
• left</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>

- right
- center
- justify
- letter-justify - Distributes the extra space between the letters.
- stretch-justify - Stretches the letters to fill in the space.

</td></tr>
<tr><td>v-text-kern</td><td>Specifies whether kerning is turned on. Default is false.</td></tr>
<tr><td>v-text-reverse</td><td>Specifies whether the layout order of rows is reversed. Default is false. This is used for vertical text layout.</td></tr>
<tr><td>v-text-spacing-mode</td><td>Specifies the mode for letter spacing. Default is tightening. This property determines whether space is removed between each letter (tightening) or added between each letter (tracking). The amount of letter spacing change is defined by the v-text-spacing property. Allowed values are:

• tightening
• tracking</td></tr>
<tr><td>v-text-spacing</td><td>Specifies the amount of spacing for text in 100ths of single line spacing. Default is 100.</td></tr>
</table>

The line (§19.1.2.12), polyline (§19.1.2.15) and curve (§19.1.2.3) elements ignore the following properties:

- top
- left
- width
- height

The following properties are not inherited by an element that references a shapetype element (§19.1.2.20) via the id attribute:

- flip
- height
- left
- margin-left
- margin-top
- position
- rotation
- top
- visibility
- width
</td></tr>
</table>

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>

- `z-index`

The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>target (Hyperlink Display Target)</td><td>

Specifies a frame or window that a URL is displayed in. Default is no value. Allowed values are:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><targetname></td><td>String containing the name of the frame or window in which to load the document.</td></tr>
<tr><td>`_blank`</td><td>Specifies that the linked document is loaded into a new blank window. This window is not named.</td></tr>
<tr><td>`_media`</td><td>Specifies that the linked document is loaded into the browser’s multimedia pane.</td></tr>
<tr><td>`_parent`</td><td>Specifies that the linked document is loaded into the immediate parent of the document containing the link.</td></tr>
<tr><td>`_search`</td><td>Specifies that the linked document is loaded into the browser's search pane.</td></tr>
<tr><td>`_self`</td><td>Specifies that the linked document is loaded into the window in which the link was clicked (the active window).</td></tr>
<tr><td>`_top`</td><td>Specifies that the linked document is loaded into the topmost window.</td></tr>
</table>

[*Example*:

```
<v:shape …
  href="http://www.openxmlformats.org"
  target="_self" … >
</v:shape>
```

end example]

The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>title (Shape Title)</td><td>

Specifies the text displayed when the mouse pointer moves over the shape. Default is no value.

[*Example*:

```
<v:shape … title="tooltip" … >
</v:shape>
```

end example]
</td></tr>
</table>

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

userdrawn (Exists In Master Slide)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the user has added the shape to a master slide. Default is `false`. Used by PresentationML.
	
	[*Example*:
	
	`<v:shape … o:userdrawn="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

userhidden (Hide Script Anchors)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a script anchor is hidden. Default is `false`. If `true`, script anchors stay hidden even if the shape is otherwise visible. A script anchor is the visual representation of a script that when displayed in an application.
	
	[*Example*:
	
	`<v:shape … o:userhidden="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

wrapcoords (Shape Bounding Polygon)	Specifies the bounding polygon that surrounds a shape. This is specified using a comma-delimited list of x and y coordinates: "x1,y1,x2,y2,x3,y3,…" This is used when text is tightly wrapped around a shape. Default is no value until the mso-wrap-mode style attribute is set to `tight` or `through`.
	
	[*Example*:
	
	`<v:shape …`
	`wrapcoords="0,0 0,200, 200,200, 200,0" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Oval) is located in §A.6.1. *end note*]

19.1.2.14 path (Shape Path)

This element defines the path that makes up the shape. This is done through a string that contains a rich set of pen movement commands. This element also describes the limo-stretch point, inscribed textbox rectangle locations and connection site locations. The limo-stretch definition and the formulas element (§19.1.2.6) allow greater designer control of how the path scales. [*Example*: They allow, for example, definition of a true rounded corner rectangle where the corners remain circular even though the rectangle is scaled anisotropically. *end example*]

Attributes	Description

arrowok (Arrowhead Display Toggle)	Specifies whether arrowheads are allowed to be displayed. This attribute overrides all other arrowhead attributes in the parent or the stroke element (§19.1.2.21). Default is `false`.
	
	[*Example*:
	
	`<v:shape style="width:50;height:50">`
	`  <v:stroke endarrow="block"/>`
	`  <v:path arrowok="true"`
	`  v="m 0,0 l 1000,0 1000,1000 e"/>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

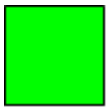
connectangles (Connection Point Connect Angles)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the angle at which curves connect to a shape's connection points. The connection angles are defined by a string consisting of angle values delimited by commas. Default is no value.
	
	[*Example*: Connections are made along the horizontal and vertical axes:
	
	`<v:path … o:connectangles="0,90,180,270" … >`
	`</v:path>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

connectlocs (Connection Points)	Specifies the location of connection points on a path. The connection points are defined by a string consisting of pairs of x and y values, delimited by commas. This is used if

Attributes	Description

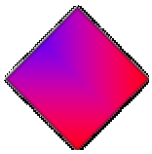
Namespace: urn:schemas-microsoft-com:office:office	connecttype is `custom`. Default is no value.
	
	[*Example*: Connection points exist at the midpoints of the sides of the square:
	
	`<v:path … v="m 0,0 l 100,0 100,100 0,100 x e"`
	`  o:connectlocs="50,0;100,50;50,100;0,50" … >`
	`</v:path>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

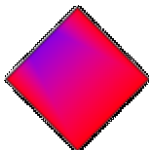
connecttype (Connection Point Type)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind ofconnection points used for attaching shapes to other shapes. Default is none. If set to `custom`, connectlocs is used. Allowed values are:
	
	[*Example*:
	
	`<v:path … o:connecttype="custom"`
	`  o:connectlocs="50,0;100,50;50,100;0,50" … >`
	`</v:path>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ConnectType simple type (§19.2.3.8).

extrusionok (Extrusion Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether an extrusion is allowed to be displayed. This attribute overrides all other extrusion attributes in the parent or the extrusion element (§19.2.2.11). Default is `true`.
	
	[*Example*:
	
	`<v:rect fillcolor="lime"`
	`  style="width:50;height:50">`
	`  <v:extrusion on="true"/>`
	`  <v:path o:extrusionok="false"/>`
	`</v:rect>`
	
	  `<v:path o:extrusionok="false"/>`
	  `<v:path o:extrusionok="true"/>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

fillok (Shape Fill Toggle)	Specifies whether a fill is allowed to be displayed. This attribute overrides all other fill attributes in the parent or fill element (§19.1.2.5). Default is `true`.
	
	[*Example*:
	
	`<v:shape style="width:50;height:50"`
	`  fillcolor="red">`
	`  <v:path fillok="false"`
	`  v="m 0,0 l 0,1000, 1000,1000, 1000,0 x e"/>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

gradientshapeok (Gradient Shape Toggle)	Specifies whether a gradient path is made up of repeated concentric paths. Default is `false`.
	
	If `true`, a gradient fill can be produced by repeated drawing of scaled versions of the path - this shall only be set if it is possible to scale the path in such a way that a fill is always contained in the original path. This controls the interpretation of the type="gradientradial" attribute of the fill element (§19.1.2.5).
	
	[*Example*: In the first case, the radial gradient is aligned irrespective of the shape's path:
	
	`<v:shape style="width:50;height:50;rotation:45"`
	`  path="m 0,0 l 0,1000, 1000,1000, 1000,0 x e">`
	`  <v:path gradientshapeok="false"/>`
	`  <v:fill type="gradientradial"`
	`  color="red" color2="blue"/>`
	`</v:shape>`
	
	 
	`gradientshapeok="false"`

Attributes	Description
	 `gradientshapeok="true"` *end example*] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

id (Unique Identifier)	Specifies a unique identifier that can be used to reference a VML object.
	
	Default is no value.
	
	[*Example*:
	
	`<v:shape … id="myShape" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

insetpenok (Inset Stroke From Path Flag)	Specifies whether the stroke can be inset from the path. If this is `false`, it overrides the insetpen attribute and prevents the stroke from being inset.
	
	[*Example*: The stroke is not inset:
	
	`<v:shape … insetpen="true">`
	`  <v:path … insetpenok="false"/>`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

limo (Limo Stretch Point)	Specifies a stretch point on the shape's edge that defines where and how a shape is allowed to be stretched by a user in a graphical editor. Default is "0,0".
	
	[*Example*:
	
	`<v:line from="20pt,20pt" to="100pt,20pt">`
	`  <v:path limo="60pt,20pt"/>`
	`</v:line>`
	
	end example]

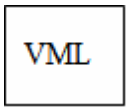
Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

shadowok (Shadow Toggle)	Specifies whether a shadow is allowed to be displayed. This attribute overrides all other shadow attributes in the parent or the shadow element (§19.1.2.18). Default is `true`.
	
	[*Example*: The shape has no shadow:
	
	`<v:shape style="width:50;height:50">`
	`  <v:path v="m 0,0 l 0,1000, 1000,1000, 1000,0 x e"`
	`  shadowok="false"/>`
	`  <v:shadow on="true"/>`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

strokeok (Stroke Toggle)	Specifies whether a stroke is displayed. This attribute overrides all other stroke attributes in the parent or the stroke element (§19.1.2.21). Default is `true`.
	
	[*Example*: The shape's red stroke is not shown:
	
	`<v:shape style="width:50;height:50"`
	`  fillcolor="blue" strokecolor="red">`
	`  <v:path v="m 0,0 l 0,1000, 1000,1000, 1000,0 x e"`
	`  strokeok="false"/>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

textboxrect (Text Box Bounding Box)	Specifies one or more text boxes inside a shape. Default is the same as the geometry's bounding box. A textbox is defined by one or more sets of numbers specifying (in order) the left, top, right, and bottom points of the rectangle. Multiple sets are delimited by a semicolon. The default value is the same dimension value as the containing rectangle. If more than one textbox is defined, the comma-delimited quadruple sets that define each textbox are separated by semicolons. Normally textboxes come in sets of 1, 2, 3, or 6 rectangles on a

Attributes	Description

	shape. The textboxrect dimensions clip any text that extends beyond its region.
	
	[*Example*: The textbox is 25% down from the top and the exclamation point is clipped:
	
	`<v:shape style="width:60;height:50">`
	`  <v:path v="m 0,0 l 0,1000, 1000,1000, 1000,0 x e"`
	`  textboxrect="0,250,850,1000"/>`
	`  <v:textbox>VML!</v:textbox>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

textpathok (Text Path Toggle)	Specifies whether a text path is displayed. Default is `false`.
	
	If `true`, this indicates that the path is an appropriate warping path for the textpath element (§19.1.2.23). Otherwise, the textpath element shall be ignored.
	
	[*Example*: The defined textpath is ignored:
	
	`<v:curve from="50,100" to="400,100"`
	`  control1="200,200" control2="300,200">`
	`  <v:path textpathok="false"/>`
	`  <v:textpath on="false" style="font:normal normal`
	`  normal 36pt Arial" string="textpath"/>`
	`</v:curve>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

v (Path Definition)	Specifies a string containing the commands that define the shape's path. This value consists of commands followed by zero or more parameters. Default is no value. The following rules apply to path strings:

Attributes	Description
	• Commas or spaces delimit parameters for each command. Both "m 0,0" and "m0 0" are acceptable. • A parameter that is omitted using commas is treated as having a value of zero. Thus, "c 10,10,0,0,25,13" and "c 10,10,,,25,13" are equivalent. • Parameterized paths are also allowed. In this case, the shape shall also have a formulas element (§19.1.2.6) with a list of formulas that are substituted into the path using the @ symbol followed by the number of the formula. The adj property of the shape contains the input parameters for these formulas. [*Example*: For example, "moveto @1@4". *end example*] The evaluations of the formulas are substituted into the appropriate positions. The @ character also serves as a delimiter. The allowed commands are given below. An asterisk (*) indicates that the command is allowed to be repeated. For the qb command, the controlpoint parameter is also allowed to be repeated.

Command	Name	Parameters	Description
m	moveto	2	Start a new sub-path at the given (x,y) coordinate.
l	lineto	2*	Draw a line from the current point to the given (x,y) coordinate which becomes the new current point. Specifying a number of coordinate pairs forms a polyline.
c	curveto	6*	Draw a cubic bézier curve from the current point to the coordinate given by the final two parameters. The control points are given by the first four parameters.
x	close	0	Close the current sub-path by drawing a straight line from the current point to the original moveto point.
e	end	0	End the current set of sub-paths. A given set of sub-paths (as delimited by end) is filled. Subsequent sets of sub-paths are filled independently and superimposed on existing ones.
t	rmoveto	2*	Start a new sub-path at a coordinate relative to the current point, cp (cpx+x, cpy+y).
r	rlineto	2*	Draw a line from the current point to the given relative coordinate (cpx+x,

Attributes	Description			
				cpy+y).
	v	rcurveto	6*	Cubic bézier curve using the given coordinate relative to the current point.
	nf	nofill	0	The current set of sub-paths (delimited by e) is not filled.
	ns	nostroke	0	The current set of sub-paths (delimited by e) is not stroked.
	ae	angleellipseto	6*	Draws a segment of an ellipse as described using these parameters. A straight line is drawn from the current point to the start point of the segment. The parameters are: center (x,y), size(w,h), start angle, end angle.
	al	angleellipse	6*	Same as angleellipseto except that there is an implied moveto the starting point of the segment.
	at	arcto	8*	A segment of the ellipse is drawn which starts at the angle defined by the start radius vector and ends at the angle defined by the end vector. A straight line is drawn from the current point to the start of the arc. The arc is always drawn in a counterclockwise direction. The parameters are: left, top, right, bottom, start(x,y), end(x,y). The first four values define the bounding box of an ellipse. The last four define two radial vectors.
	ar	arc	8*	Same as arcto except there is an implied moveto the start point of the arc.
	wa	clockwisearcto	8*	Same as arcto but the arc is drawn in a clockwise direction.
	wr	clockwisearc	8*	Same as arc but the arc is drawn in a clockwise direction

Attributes	Description
	<table><tr><td>qx</td><td>ellipticalqaudrantx</td><td>2*</td><td>A quarter ellipse is drawn from the current point to the given end point. The elliptical segment is initially tangential to a line parallel to the x-axis. (i.e. the segment starts out horizontal). The parameters are: end(x,y).</td></tr><tr><td>qy</td><td>ellipticalquadranty</td><td>2*</td><td>Same as ellipticalquadrantx except that the elliptical segment is initially tangential to a line parallel to the y-axis (i.e. the segment starts out vertical).</td></tr><tr><td>qb</td><td>quadraticbezier</td><td>2+2*</td><td>Defines one or more quadratic bézier curves by means of control points and an end point. Intermediate (on-curve) points are obtained by interpolation between successive control points as in the OpenType font specification. The sub-path need not be started in which case the sub-path is closed. In this case the last point of the sub-path defines the start point of the quadratic bézier. The parameters are: controlpoint(x,y)*, end(x,y).</td></tr></table> The possible values for this attribute are defined by the W3C XML Schema string datatype.

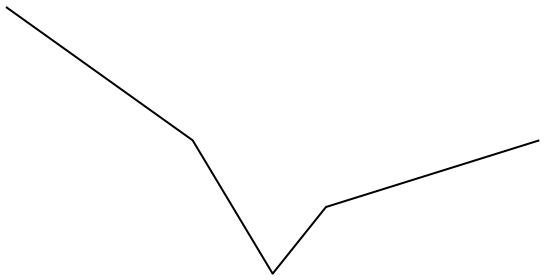
[*Note*: The W3C XML Schema definition of this element's content model (CT_Path) is located in §A.6.1. *end note*]

19.1.2.15 polyline (Multiple Path Line)

This element defines shapes made up of connected line segments.

[*Example*:

```
<v:polyline
  points="50pt,0pt 120pt,50pt 150pt,100pt 170pt,75pt 250pt,50pt">
</v:polyline>
```

end example]

Attributes	Description

allowincell (Allow in Table Cell)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can be placed in a table. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:allowincell="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

allowoverlap (Allow Shape Overlap)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can overlap another shape. Default is `true`. If `false`, the shape shifts left or right so as not to overlap another shape, similar to the behavior of the HTML float attribute.
	
	[*Example*:
	
	`<v:shape … o:allowoverlap="false" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

alt (Alternate Text)	Specifies alternative text describing the graphical object. This text should provide a brief description of the shape for use by accessibility tools. Default is no value.
	
	[*Example*: The alt text describes the basic shape:
	
	`<v:shape … fillcolor="red"`
	`  alt="Red rectangle">`
	`</v:shape>`
	
	The alt text describes the contents of a shape displaying an image:

Attributes	Description

	`<v:shape … alt="Picture of a sunset">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderbottomcolor (Bottom Border Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the bottom border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderbottomcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderleftcolor (Border Left Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the left border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderleftcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderrightcolor (Border Right Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the right border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderrightcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

bordertopcolor (Border Top Color) Namespace: urn:schemas-	Specifies the top border color of an inline shape. Default is no value. [*Example*: `<v:shape … o:bordertopcolor="red" … >`

Attributes	Description
microsoft-com:office:office	`</v:shape>` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

bullet (Graphical Bullet)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the shape is a graphical bullet. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:bullet="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

button (Button Behavior Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape exhibits button press behavior on click. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:button="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

bwmode (Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies how a shape renders for black-and-white output devices. When a shape is printed on a black-and-white printer or displayed in a black-and-white view in an application, several options are possible. Default is `auto`, which uses o:bwnormal for normal black-and-white rendering and o:bwpure for pure black-and-white rendering.
	
	bwnormal and bwpure are subordinate to bwmode. If bwmode is "auto" then the value for bwnormal or bwpure is used depending on what the output format is. An application can define for itself what, if any, difference there is between normal B&W and pure B&W. [*Example*: Normal B&W might allow greyscale and pure B&W might not. *end example*]
	[*Example*: This shape renders in grayscale in a black-and-white environment:
	
	`<v:shape … o:bwmode="grayscale" … >`
	`</v:shape>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

bwnormal (Normal Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the black-and-white mode for normal black-and-white output devices. Default is `auto`.
	
	[*Example*: This shape renders in a pale grayscale in a normal black-and-white environment:
	
	`<v:shape … o:bwmode="auto" o:bwnormal="lightgrayscale" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

bwpure (Pure Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the black-and-white mode for pure black-and-white output devices. Default is `auto`.
	
	[*Example*: This shape renders in high contrast when in a pure black-and-white environment:
	
	`<v:shape … o:bwmode="auto" o:bwpure="highcontrast" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

chromakey (Image Transparency Color)	Specifies a color value that is transparent and show anything behind the shape. Default is no value.
	
	[*Example*:
	
	`<v:image … chromakey="white" …>`
	`</v:image>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

class (CSS Reference)	Specifies a reference to the definition of a CSS style. Default is no value. [*Example*: The snippets below are equivalent: …

Attributes	Description

	`.narrowstyle {width:50;height:100}`
	`…`
	`<v:shape … class="narrowstyle"`
	`  style="top:1;left:1">`
	`</v:shape>`
	
	`<v:shape … style="top:1;left:1;`
	`  width:50;height:100">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

clip (Clipping Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the clipping region is active. This is used in conjunction with the clippath (§19.2.2.3) element to create a clipping region.
	
	[*Example*:
	
	`<v:shape … o:clip="true">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

cliptowrap (Clip to Wrapping Polygon)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the clipping region for the shape aligns with the wrapping polygon that tightly surrounds the entire shape (essentially, that the shape shall not be drawn beyond its wrapping polygon's extents – if it does, the shape shall be clipped). Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:cliptowrap="true">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

connectortype (Shape Connector Type)
 	
Namespace: urn:schemas-
	Specifies the kind of connector used for joining shapes. Default is `straight`.
	
	[*Example*:
	
	`<v:shape … o:connectortype="elbow" … >`
	`</v:shape>`

Attributes	Description
microsoft-com:office:office	*end example*] The possible values for this attribute are defined by the ST_ConnectorType simple type (§19.2.3.7).

coordorigin (Coordinate Space Origin)	Specifies the coordinate of the top left corner of the shape's coordinate space. This determines the position of the (0,0) coordinate space origin within the shape's bounding box. Default is "0,0", which places the (0,0) origin at the top left corner of the bounding box.
	
	This affects shape properties that specify coordinate positions, such as the path attribute. Thus a path can be defined against a generic (0,0) origin and the coordorigin value translates the entire path within the shape's bounding space.
	
	[*Example*: The horizontal and vertical coordinate space ranges from -100 to +100 because the coordinate space (coordsize) is 200 by 200 and the top left coordinate is (-100,-100). The (0,0) origin lies at the center of the shape's bounding box, as evidenced by the position of the shape's path within the coordinate space:
	
	`<v:shape … coordsize="200,200"`
	`  coordorigin="-100,-100"`
	`  path="m 0,0 l 0,50, 50,50, 50,0 x e">`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

coordsize (Coordinate Space Size)	Specifies the size of the shape's coordinate space in coordinate units. Default is "1000,1000".
	
	The physical size of a coordinate unit length is determined by both the size of the coordinate space (coordsize) and the size of the shape (style width and height). The coordsize attribute defines the number of horizontal and vertical subdivisions into which the shape's bounding box is divided. The combination of coordsize and style width/height effective scales the shape anisotropically.
	
	[*Example*: The path is 50 units wide and tall, which is 25% of the size of the coordinate space:
	
	`<v:shape … coordsize="200,200"`
	`  coordorigin="-100,-100"`

Attributes	Description

	`path="m 0,0 l 0,50, 50,50, 50,0 x e">`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

dgmlayout (Diagram Node Layout Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of automatic layout to apply to the diagram node. This is only meaningful if the shape is a node in an organization chart, which is denoted by the `orgchart` value of the editas attribute of the group element.
	[*Example*:
	
	`<v:shape … dgmlayout="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).

dgmlayoutmru (Diagram Node Recent Layout Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of automatic layout most recently used on the child elements of the diagram node. This is only meaningful if the shape is a node in anorganization chart, which is denoted by the `orgchart` value of the editas attribute of the group element.
	
	[*Example*:
	
	`<v:shape … dgmlayout="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).

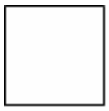
dgmnodekind (Diagram Node Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies an optional, application-defined parameter that is intended to be used by the application to tag different types of nodes in a diagram.
	
	[*Example*:
	
	`<v:shape … dgmnodekind="1">`
	`</v:shape>`

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema integer datatype.

doubleclicknotify (Double-click Notification Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that an event message is sent when a shape is double-clicked. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:doubleclicknotify="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

fillcolor (Fill Color)	Specifies the color to use for the fill. Default is `white`. If the fill element (§19.1.2.5) is present, its color attribute takes precedence. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*: This shape is red if its fill is visible:
	
	`<v:shape … fillcolor="red" … >`
	`</v:shape>`
	
	This is equivalent to:
	
	`<v:shape … fillcolor="#ff0000" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

filled (Shape Fill Toggle)	Specifies whether the closed path is filled. Default is `true`. This attribute is overridden by the fill on attribute.
	
	[*Example*:
	
	`<v:shape … filled="f"`
	`  fillcolor="red" …>`
	`</v:shape>`

Attributes	Description
	 *end example*] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

forcedash (Force Dashed Outline)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a dashed outline is used to draw a shape when a shape has no line or fill. Default is `false`.
	
	Used by PresentationML placeholders to draw a dashed outline when there is no line and no fill for a shape.
	
	[*Example*:
	
	`<v:shape … o:forcedash="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hr (Horizontal Rule Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that a shape is a horizontal rule. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hr="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hralign (Horizontal Rule Alignment)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the alignment of a horizontal rule. Default is `left`.
	
	[*Example*:
	
	`<v:shape … o:hralign="center" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_HrAlign simple type (§19.2.3.16).

Attributes	Description

href (Hyperlink Target)	Specifies a hyperlink URL target for the shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … href="http://www.openxmlformats.org" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

hrnoshade (Horizontal Rule 3D Shading Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the horizontal rule does not have 3-D shading. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hrnoshade="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hrpct (Horizontal Rule Length Percentage)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the length of a horizontal rule as a percentage of page width. Default is 0.
	
	[*Example*:
	
	`<v:shape … o:hrpct="85" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema float datatype.

hrstd (Horizontal Rule Standard Display Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape is displayed as a standard horizontal rule. Only applies if hr is `true`. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hrstd="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

id (Unique	Specifies a unique identifier that can be used to reference a VML object.

Attributes	Description

Identifier)	Default is no value.
	
	[*Example*:
	
	`<v:shape … id="myShape" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

insetmode (Text Inset Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the application calculates the internal text margin instead of using the inset attribute. Default is `custom`. This attribute is only meaningful for text boxes.
	
	[*Example*:
	
	`<v:shape … o:insetmode="auto" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_InsetMode simple type (§19.2.3.17).

insetpen (Inset Border From Path)	Specifies that the border shall be displayed inside of the path defining the shape, rather than along the path (the default border placement), or outside of the path as might be done with an image.
	
	[*Example*:
	
	`<v:shape … insetpen="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

ole (Embedded Object Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the shape is an embedded object. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:ole="true" … >`
	`</v:shape>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

oleicon (Embedded Object Icon Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether an embedded object is displayed as an icon. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oleicon="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

oned (Shape Handle Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the extra handles of a shape are hidden. If `true`, hides all shape handles except the top left and bottom right; that is, the same handles that are used for a straight line segment. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oned="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

opacity (Fill Color Opacity)	Specifies the opacity of the primary fill color. Default is 1.0. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. [*Example*: A value of "52429f" represents 52429/65536 or 0.8. *end example*]
	
	[*Example*: The red color is 25% opaque:
	
	`<v:fill type="gradient" color="red"`
	`color2="blue" opacity=".25">`
	`</v:fill>`
	
	  `opacity="1"`
	
	  `opacity=".25"`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.
points (Points for Compound Line)	Specifies a set of straight line segments that are composed of a series of pairs of points. Default is "0,0 10,10". Points are specified in the coordinate system of the parent element. If the parent is not a VML element, the default unit is a pixel. Allowed units are in, cm, mm, pt, pc and px. While commas are not required, they should be used for easier readability. See above for an example. The possible values for this attribute are defined by the W3C XML Schema string datatype.

preferrelative (Relative Resize Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the original size of an object is saved after reformatting. Default is `false`. If `true`, the original size of the object is stored and all resizing is based on a percentage of that original size. Otherwise, each resizing resets the scale to 100%.
	
	[*Example*:
	
	`<v:shape … o:preferrelative="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

print (Print Toggle)	Specifies whether the shape is printed. Default is `true`.
	
	[*Example*:
	
	`<v:shape … print="false" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

regroupid (Regroup ID)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies a previous group for a shape. An ID number is used to identify groups of shapes that are no longer grouped. This allows shapes to be regrouped programmatically.
	
	[*Example*: The shape was part of a group identified by the ID 040754:
	
	`<v:shape … o:regroupid="040754" … >`
	`</v:shape>`

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema integer datatype.
spid (Optional String) Namespace: urn:schemas-microsoft-com:office:office	Specifies an optional string that an application can use to Identify the particular shape. Default is no value. The possible values for this attribute are defined by the W3C XML Schema string datatype.
spt (Optional Number) Namespace: urn:schemas-microsoft-com:office:office	Specifies an optional number that an application can use to associate the particular shape with a defined shape type. Default is 0. The possible values for this attribute are defined by the W3C XML Schema float datatype.

strokecolor (Shape Stroke Color)	Specifies the primary color of the brush to use to stroke the path of the shape. Default is `black`. The color attribute of the stroke element (§19.1.2.21) overrides this. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*:
	
	`<v:shape … strokecolor="red" …>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

stroked (Shape Stroke Toggle)	Specifies whether the path defining the shape is stroked with a solid line. The stroke element (§19.1.2.21) defines other strokes. The on attribute of the stroke element overrides this attribute. Default is `true`. [*Example*: `<v:shape … fillcolor="red"`

Attributes	Description

	`stroked="false" strokecolor="blue"…>`
	`</v:shape>`
	 
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

strokeweight (Shape Stroke Weight)	Specifies the width of the brush to use to stroke the path. Default is 1 point. If a number is given without units, the emu is used. The weight attribute of the stroke element (§19.1.2.21) overrides this attribute.
	
	[*Example*:
	
	`<v:shape … strokeweight="3pt" … >`
	`</v:shape>`
	 
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

style (Shape Styling Properties)	Specifies the CSS2 styling properties of the shape. The CSS2 (Cascading Style Sheets, Level 2) specification, a Recommendation of the World Wide Web Consortium, is available here: http://www.w3.org/TR/REC-CSS2.
	
	This attribute uses a semi-colon delimited list of “name:value” pairs, the syntax defined by section 4.1.8 of the CSS2 specification for grouped declarations (without the surrounding braces). The following tables define the allowed properties and the VML treatment of each.
	
	[*Example*:
	`<v:shape … style=’position:absolute;width:100pt;height:50pt’ …`
	`>`
	`</v:shape>`
	end example]
	
	Property | **Description**

Attributes	Description	
	`flip`	Specifies that the orientation of a shape is flipped. Default is no value. Allowed values are: • x - Flip along the y-axis, reversing the x-coordinates. • y - Flip along the x-axis, reversing the y-coordinates. • xy - Flip along both the y- and x-axis. • yx - Flip along both the x- and y-axis.
	`height`	Specifies the height of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`left`	Specifies the position of the left of the containing block of the shape relative to the element left of it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-bottom`	Specifies the position of the bottom of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.

Attributes	Description	
	`margin-left`	Specifies the position of the left of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-right`	Specifies the position of the right of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-top`	Specifies the position of the top of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`mso-position-horizontal`	Specifies the horizontal positioning data for objects in WordprocessingML documents. Default is `absolute`. Allowed values are: • `absolute` • `left` • `center` • `right`

Attributes	Description	
		• inside • outside
	mso-position-horizontal-relative	Specifies relative horizontal position data for objects in WordprocessingML documents. This modifies the mso-position-horizontal property. Default is text. Allowed values are: • margin • page • text • char
	mso-position-vertical	Specifies the vertical positioning data for objects in WordprocessingML documents. Default is absolute. Allowed values are: • absolute • top • center • bottom • inside • outside
	mso-position-vertical-relative	Specifies relative vertical position data for objects in WordprocessingML documents. This modifies the mso-position-vertical property. Default is text. Allowed values are: • margin • page • text • line
	mso-wrap-distance-bottom	Specifies the distance from the bottom of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	mso-wrap-distance-left	Specifies the distance from the left side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	mso-wrap-	Specifies the distance from the right side of the shape to the text

Attributes	Description
	`distance-right` — that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	`mso-wrap-distance-top` — Specifies the distance from the top of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	`mso-wrap-edited` — Specifies whether the wrap coordinates were customized by the user. If the wrap coordinates are generated by an editor, this property is `true`; otherwise they were customized by a user. Default is `false`.
	`mso-wrap-style` — Specifies the wrapping mode for text in shapes in WordprocessingML documents. Default is `square`. Allowed values are: • `square` - Wraps text inside the shape in a square. • `none` - Text does not wrap.
	`position` — Specifies the kind of positioning used to place an element. Default is `static`. When the element is contained inside a group, this property shall be `absolute`. Allowed values are: • `static` - The element is positioned according to the normal flow of the page. The `top` and `left` properties are ignored. If the object is anchored inline, this value is used. • `absolute` - The element is positioned relative to the parent, using the `top` and `left` properties. • `relative` - The element is positioned according to the normal flow of the page, but the `top` and `left` properties are used. The overlap of overlapping elements is governed by the `z-index` property.
	`rotation` — Specifies the angle that a shape is rotated, in degrees. Default is 0. Positive angles are clockwise.
	`top` — Specifies the position of the top of the containing block of the shape relative to the element above it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page.

Attributes	Description
	• <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
`visibility`	Specifies whether a shape is displayed. Only `inherit` and `hidden` are used; any other values are mapped to `inherit`. Default is `inherit`. Allowed values are: • `hidden` - The shape is not visible, but is still part of the flow of the objects in the browser. Mouse events are not processed. • `inherit` - The visibility state is inherited from the parent of the shape.
`width`	Specifies the width of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
`z-index`	Specifies the display order of overlapping shapes. Default is 0. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Uses the order that the shapes appear in the page, bottom to top. • <order>- A number that represents the stacking precedence. Shapes with higher numbers are placed on top of those with lower numbers. Negative numbers are allowed.

The following properties are only used by the textbox element (§19.1.2.22):

Property	Description
`direction`	Specifies the direction of the text in the textbox. Default is `ltr`. This property is superceded by the `mso-direction-alt` property if that is specified. Allowed values are:

Attributes	Description
	• `ltr` - Text is displayed left-to-right. • `rtl` - Text is displayed right-to-left.
`layout-flow`	Determines the flow of the text layout in a textbox. Default is `horizontal`. Allowed values are: • `horizontal` - Text is displayed horizontally. • `vertical` - Text is displayed vertically. • `vertical-ideographic` - Ideographic text is displayed vertically. • `horizontal-ideographic` - Ideographic text is displayed horizontally.
`mso-direction-alt`	Specifies an alternate direction for text in textboxes. Overrides the `direction` property. The only allowed value is `context`.
`mso-fit-shape-to-text`	Specifies whether the shape stretches to fit the text in the textbox. Default is `false`.
`mso-fit-text-to-shape`	Specifies whether the text stretches to fit the textbox. Default is `false`.
`mso-layout-flow-alt`	Specifies the alternate layout flow for text in textboxes. This property is used instead of `layout-flow` when the layout flow is from bottom to top for non-ideographic languages. Its only value is `bottom-to-top`.
`mso-next-textbox`	Specifies the ID of the next textbox in a series. Used to keep track of a set of linked textboxes. Default is no value.
`mso-rotate`	Specifies a specific rotation value for text in a textbox. Default is 0. Allowed values are: • 0 • 90 • 180 • -90
`mso-text-scale`	Specifies the scaling factor for fitting text to shapes. Default is 0. This property is only used if `mso-fit-text-to-shape` is `true`.
`v-text-anchor`	Specifies the vertical anchoring of text in a textbox. Default is `top`. The alignment of a text anchor only becomes evident if `mso-fit-text-to-shape` is `false`. This property is different from the `vertical-align` CSS property, which is used for ideographic languages. Allowed values are:

Attributes	Description
	<table><tr><td></td><td>• top • middle • bottom • top-center • middle-center • bottom-center • top-baseline • bottom-baseline • top-center-baseline • bottom-center-baseline</td></tr></table> The following properties are only used by the textpath element (§19.1.2.23): <table><tr><th>Property</th><th>Description</th></tr><tr><td>font</td><td>Specifies a compound value of font settings. Default is no value. The values are the same as those of the CSS font property. The order of definitions in the string is: font-style, font-variant, font-weight, font-size, line-height, font-family.</td></tr><tr><td>font-family</td><td>Specifies the family of the font. Default is no value. The values are the same as those of the CSS font-family property.</td></tr><tr><td>font-size</td><td>Specifies the size of the font. Default is no value. The font size is defined in points. The values are the same as those of the CSS font-size property.</td></tr><tr><td>font-style</td><td>Specifies the amount of slant for a font. Default is normal. The values are the same as those of the CSS font-style property. Allowed values are: • normal • italic • oblique - Treated the same as italic.</td></tr><tr><td>font-variant</td><td>Specifies the variant style of a font. Default is normal. The values are the same as those of the CSS font-variant property. Allowed values are: • normal • small-caps</td></tr><tr><td>font-weight</td><td>Specifies the thickness of the letters of the font. Default is normal. The values are the same as those of the CSS font-weight property. Allowed values are: <table><tr><th>Value</th><th>Description</th></tr></table></td></tr></table>

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>
<table>
<tr><td>normal
lighter
100
200
300
400</td><td>Treated as non-bold.</td></tr>
<tr><td>bold
bolder
500
600
700
800
900</td><td>Treated as bold.</td></tr>
</table>
</td></tr>
<tr><td>mso-text-shadow</td><td>Specifies whether a shadow is applied to the text on a text path. Default is false.</td></tr>
<tr><td>text-decoration</td><td>Specifies the style of text decoration. Default is none. The values are the same as those of the CSS text-decoration property. Allowed values are:
• none
• underline
• overline
• line-through
• blink</td></tr>
<tr><td>v-rotate-letters</td><td>Specifies whether the letters of the text are rotated counterclockwise by 90 degrees. Default is false.</td></tr>
<tr><td>v-same-letter-heights</td><td>Specifies whether all letters are the same height regardless of initial case. If true, the lowercase letters are stretched to the height of the uppercase letters. Default is false.</td></tr>
<tr><td>v-text-align</td><td>Specifies the alignment of text. Default is left. Allowed values are:
• left</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>

- right
- center
- justify
- letter-justify - Distributes the extra space between the letters.
- stretch-justify - Stretches the letters to fill in the space.

</td></tr>
<tr><td>v-text-kern</td><td>Specifies whether kerning is turned on. Default is false.</td></tr>
<tr><td>v-text-reverse</td><td>Specifies whether the layout order of rows is reversed. Default is false. This is used for vertical text layout.</td></tr>
<tr><td>v-text-spacing-mode</td><td>Specifies the mode for letter spacing. Default is tightening. This property determines whether space is removed between each letter (tightening) or added between each letter (tracking). The amount of letter spacing change is defined by the v-text-spacing property. Allowed values are:

• tightening
• tracking</td></tr>
<tr><td>v-text-spacing</td><td>Specifies the amount of spacing for text in 100ths of single line spacing. Default is 100.</td></tr>
</table>

The line (§19.1.2.12), polyline (§19.1.2.15) and curve (§19.1.2.3) elements ignore the following properties:

- top
- left
- width
- height

The following properties are not inherited by an element that references a shapetype element (§19.1.2.20) via the id attribute:

- flip
- height
- left
- margin-left
- margin-top
- position
- rotation
- top
- visibility
- width
</td></tr>
</table>

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>

- `z-index`

The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>target (Hyperlink Display Target)</td><td>

Specifies a frame or window that a URL is displayed in. Default is no value. Allowed values are:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><targetname></td><td>String containing the name of the frame or window in which to load the document.</td></tr>
<tr><td>`_blank`</td><td>Specifies that the linked document is loaded into a new blank window. This window is not named.</td></tr>
<tr><td>`_media`</td><td>Specifies that the linked document is loaded into the browser’s multimedia pane.</td></tr>
<tr><td>`_parent`</td><td>Specifies that the linked document is loaded into the immediate parent of the document containing the link.</td></tr>
<tr><td>`_search`</td><td>Specifies that the linked document is loaded into the browser's search pane.</td></tr>
<tr><td>`_self`</td><td>Specifies that the linked document is loaded into the window in which the link was clicked (the active window).</td></tr>
<tr><td>`_top`</td><td>Specifies that the linked document is loaded into the topmost window.</td></tr>
</table>

[*Example*:

```
<v:shape …
  href="http://www.openxmlformats.org"
  target="_self" … >
</v:shape>
```

end example]

The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>title (Shape Title)</td><td>

Specifies the text displayed when the mouse pointer moves over the shape. Default is no value.

[*Example*:

```
<v:shape … title="tooltip" … >
</v:shape>
```

end example]
</td></tr>
</table>

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

userdrawn (Exists In Master Slide)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the user has added the shape to a master slide. Default is `false`. Used by PresentationML.
	
	[*Example*:
	
	`<v:shape … o:userdrawn="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

userhidden (Hide Script Anchors)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a script anchor is hidden. Default is `false`. If `true`, script anchors stay hidden even if the shape is otherwise visible. A script anchor is the visual representation of a script that when displayed in an application.
	
	[*Example*:
	
	`<v:shape … o:userhidden="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

wrapcoords (Shape Bounding Polygon)	Specifies the bounding polygon that surrounds a shape. This is specified using a comma-delimited list of x and y coordinates: "x1,y1,x2,y2,x3,y3,…" This is used when text is tightly wrapped around a shape. Default is no value until the mso-wrap-mode style attribute is set to `tight` or `through`.
	
	[*Example*:
	
	`<v:shape …`
	`wrapcoords="0,0 0,200, 200,200, 200,0" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

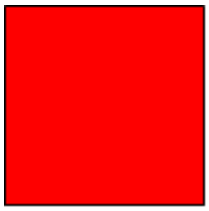
[*Note*: The W3C XML Schema definition of this element's content model (CT_PolyLine) is located in §A.6.1. *end note*]

19.1.2.16 rect (Rectangle)

This element is used to draw a simple rectangle. The CSS2 style content width and height define the width and height of the rectangle.

[*Example*:

```
<v:rect fillcolor="red"
  style="position:relative;top:0;left:0;width:100;height:100">
</v:rect>
```



end example]

Attributes	Description

allowincell (Allow in Table Cell)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can be placed in a table. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:allowincell="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

allowoverlap (Allow Shape Overlap)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can overlap another shape. Default is `true`. If `false`, the shape shifts left or right so as not to overlap another shape, similar to the behavior of the HTML float attribute.
	
	[*Example*:
	
	`<v:shape … o:allowoverlap="false" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

Attributes	Description

alt (Alternate Text)	Specifies alternative text describing the graphical object. This text should provide a brief description of the shape for use by accessibility tools. Default is no value.
	
	[*Example*: The alt text describes the basic shape:
	
	`<v:shape … fillcolor="red"`
	`  alt="Red rectangle">`
	`</v:shape>`
	
	The alt text describes the contents of a shape displaying an image:
	
	`<v:shape … alt="Picture of a sunset">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderbottomcolor (Bottom Border Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the bottom border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderbottomcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderleftcolor (Border Left Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the left border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderleftcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderrightcolor (Border Right Color) Namespace: urn:schemas-	Specifies the right border color of an inline shape. Default is no value. [*Example*: `<v:shape … o:borderrightcolor="red" … >`

Attributes	Description
microsoft-com:office:office	`</v:shape>` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

bordertopcolor (Border Top Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the top border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:bordertopcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

bullet (Graphical Bullet)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the shape is a graphical bullet. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:bullet="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

button (Button Behavior Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape exhibits button press behavior on click. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:button="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

bwmode (Black-and-White Mode) Namespace: urn:schemas-microsoft-	Specifies how a shape renders for black-and-white output devices. When a shape is printed on a black-and-white printer or displayed in a black-and-white view in an application, several options are possible. Default is `auto`, which uses o:bwnormal for normal black-and-white rendering and o:bwpure for pure black-and-white rendering. bwnormal and bwpure are subordinate to bwmode. If bwmode is "auto" then the value

Attributes	Description

com:office:office	for bwnormal or bwpure is used depending on what the output format is. An application can define for itself what, if any, difference there is between normal B&W and pure B&W. [*Example*: Normal B&W might allow greyscale and pure B&W might not. *end example*]
	[*Example*: This shape renders in grayscale in a black-and-white environment:
	
	`<v:shape … o:bwmode="grayscale" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

bwnormal (Normal Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the black-and-white mode for normal black-and-white output devices. Default is `auto`.
	
	[*Example*: This shape renders in a pale grayscale in a normal black-and-white environment:
	
	`<v:shape … o:bwmode="auto" o:bwnormal="lightgrayscale" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

bwpure (Pure Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the black-and-white mode for pure black-and-white output devices. Default is `auto`.
	
	[*Example*: This shape renders in high contrast when in a pure black-and-white environment:
	
	`<v:shape … o:bwmode="auto" o:bwpure="highcontrast" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

chromakey (Image Transparency Color)	Specifies a color value that is transparent and show anything behind the shape. Default is no value. [*Example*: `<v:image … chromakey="white" …>`

Attributes	Description
	`</v:image>` *end example*] The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

class (CSS Reference)	Specifies a reference to the definition of a CSS style. Default is no value.
	
	[*Example*: The snippets below are equivalent:
	
	`…`
	`.narrowstyle {width:50;height:100}`
	`…`
	`<v:shape … class="narrowstyle"`
	`  style="top:1;left:1">`
	`</v:shape>`
	
	`<v:shape … style="top:1;left:1;`
	`  width:50;height:100">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

clip (Clipping Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the clipping region is active. This is used in conjunction with the clippath (§19.2.2.3) element to create a clipping region.
	
	[*Example*:
	
	`<v:shape … o:clip="true">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

cliptowrap (Clip to Wrapping Polygon)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the clipping region for the shape aligns with the wrapping polygon that tightly surrounds the entire shape (essentially, that the shape shall not be drawn beyond its wrapping polygon's extents – if it does, the shape shall be clipped). Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:cliptowrap="true">`
	`</v:shape>`

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

connectortype (Shape Connector Type)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of connector used for joining shapes. Default is `straight`.
	
	[*Example*:
	
	`<v:shape … o:connectortype="elbow" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ConnectorType simple type (§19.2.3.7).

coordorigin (Coordinate Space Origin)	Specifies the coordinate of the top left corner of the shape's coordinate space. This determines the position of the (0,0) coordinate space origin within the shape's bounding box. Default is "0,0", which places the (0,0) origin at the top left corner of the bounding box.
	
	This affects shape properties that specify coordinate positions, such as the path attribute. Thus a path can be defined against a generic (0,0) origin and the coordorigin value translates the entire path within the shape's bounding space.
	
	[*Example*: The horizontal and vertical coordinate space ranges from -100 to +100 because the coordinate space (coordsize) is 200 by 200 and the top left coordinate is (-100,-100). The (0,0) origin lies at the center of the shape's bounding box, as evidenced by the position of the shape's path within the coordinate space:
	
	`<v:shape … coordsize="200,200"`
	`  coordorigin="-100,-100"`
	`  path="m 0,0 l 0,50, 50,50, 50,0 x e">`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

coordsize (Coordinate Space	Specifies the size of the shape's coordinate space in coordinate units. Default is "1000,1000".

Attributes	Description

Size)	The physical size of a coordinate unit length is determined by both the size of the coordinate space (coordsize) and the size of the shape (style width and height). The coordsize attribute defines the number of horizontal and vertical subdivisions into which the shape's bounding box is divided. The combination of coordsize and style width/height effective scales the shape anisotropically.
	
	[*Example*: The path is 50 units wide and tall, which is 25% of the size of the coordinate space:
	
	`<v:shape … coordsize="200,200"`
	`  coordorigin="-100,-100"`
	`  path="m 0,0 l 0,50, 50,50, 50,0 x e">`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

dgmlayout (Diagram Node Layout Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of automatic layout to apply to the diagram node. This is only meaningful if the shape is a node in an organization chart, which is denoted by the `orgchart` value of the editas attribute of the group element.
	[*Example*:
	
	`<v:shape … dgmlayout="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).

dgmlayoutmru (Diagram Node Recent Layout Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of automatic layout most recently used on the child elements of the diagram node. This is only meaningful if the shape is a node in anorganization chart, which is denoted by the `orgchart` value of the editas attribute of the group element.
	
	[*Example*:
	
	`<v:shape … dgmlayout="1">`
	`</v:shape>`
	
	end example]

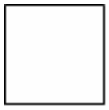
Attributes	Description
	The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).

dgmnodekind (Diagram Node Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies an optional, application-defined parameter that is intended to be used by the application to tag different types of nodes in a diagram.
	
	[*Example*:
	
	`<v:shape … dgmnodekind="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

doubleclicknotify (Double-click Notification Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that an event message is sent when a shape is double-clicked. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:doubleclicknotify="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

fillcolor (Fill Color)	Specifies the color to use for the fill. Default is `white`. If the fill element (§19.1.2.5) is present, its color attribute takes precedence. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*: This shape is red if its fill is visible:
	
	`<v:shape … fillcolor="red" … >`
	`</v:shape>`
	
	This is equivalent to:
	
	`<v:shape … fillcolor="#ff0000" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type

Attributes	Description
	(§20.1.2.3).

filled (Shape Fill Toggle)	Specifies whether the closed path is filled. Default is `true`. This attribute is overridden by the fill on attribute.
	
	[*Example*:
	
	`<v:shape … filled="f"`
	`  fillcolor="red" …>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

forcedash (Force Dashed Outline)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a dashed outline is used to draw a shape when a shape has no line or fill. Default is `false`.
	
	Used by PresentationML placeholders to draw a dashed outline when there is no line and no fill for a shape.
	
	[*Example*:
	
	`<v:shape … o:forcedash="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hr (Horizontal Rule Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that a shape is a horizontal rule. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hr="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hralign (Horizontal	Specifies the alignment of a horizontal rule. Default is `left`.

Attributes	Description

Rule Alignment)
 	
Namespace: urn:schemas-microsoft-com:office:office
	[*Example*:
	
	`<v:shape … o:hralign="center" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_HrAlign simple type (§19.2.3.16).

href (Hyperlink Target)	Specifies a hyperlink URL target for the shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … href="http://www.openxmlformats.org" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

hrnoshade (Horizontal Rule 3D Shading Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the horizontal rule does not have 3-D shading. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hrnoshade="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hrpct (Horizontal Rule Length Percentage)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the length of a horizontal rule as a percentage of page width. Default is 0.
	
	[*Example*:
	
	`<v:shape … o:hrpct="85" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema float datatype.

hrstd (Horizontal Rule Standard Display Toggle)	Specifies whether a shape is displayed as a standard horizontal rule. Only applies if hr is `true`. Default is `false`.

Attributes	Description

Namespace: urn:schemas-microsoft-com:office:office	[*Example*:
	
	`<v:shape … o:hrstd="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

id (Unique Identifier)	Specifies a unique identifier that can be used to reference a VML object.
	
	Default is no value.
	
	[*Example*:
	
	`<v:shape … id="myShape" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

insetmode (Text Inset Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the application calculates the internal text margin instead of using the inset attribute. Default is `custom`. This attribute is only meaningful for text boxes.
	
	[*Example*:
	
	`<v:shape … o:insetmode="auto" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_InsetMode simple type (§19.2.3.17).

insetpen (Inset Border From Path)	Specifies that the border shall be displayed inside of the path defining the shape, rather than along the path (the default border placement), or outside of the path as might be done with an image.
	
	[*Example*:
	
	`<v:shape … insetpen="true" … >`
	`</v:shape>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

ole (Embedded Object Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the shape is an embedded object. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:ole="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

oleicon (Embedded Object Icon Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether an embedded object is displayed as an icon. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oleicon="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

oned (Shape Handle Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the extra handles of a shape are hidden. If `true`, hides all shape handles except the top left and bottom right; that is, the same handles that are used for a straight line segment. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oned="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

opacity (Fill Color Opacity)	Specifies the opacity of the primary fill color. Default is 1.0. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. [*Example*: A value of "52429f" represents 52429/65536 or 0.8. *end example*]
	
	[*Example*: The red color is 25% opaque:
	
	`<v:fill type="gradient" color="red"`
	`color2="blue" opacity=".25">`

Attributes	Description
	`</v:fill>`  `opacity="1"`  `opacity=".25"` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

preferrelative (Relative Resize Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the original size of an object is saved after reformatting. Default is `false`. If `true`, the original size of the object is stored and all resizing is based on a percentage of that original size. Otherwise, each resizing resets the scale to 100%.
	
	[*Example*:
	
	`<v:shape … o:preferrelative="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

print (Print Toggle)	Specifies whether the shape is printed. Default is `true`.
	
	[*Example*:
	
	`<v:shape … print="false" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

regroupid (Regroup ID)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies a previous group for a shape. An ID number is used to identify groups of shapes that are no longer grouped. This allows shapes to be regrouped programmatically.
	
	[*Example*: The shape was part of a group identified by the ID 040754:
	
	`<v:shape … o:regroupid="040754" … >`
	`</v:shape>`

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema integer datatype.
spid (Optional String) Namespace: urn:schemas-microsoft-com:office:office	Specifies an optional string that an application can use to Identify the particular shape. Default is no value. The possible values for this attribute are defined by the W3C XML Schema string datatype.
spt (Optional Number) Namespace: urn:schemas-microsoft-com:office:office	Specifies an optional number that an application can use to associate the particular shape with a defined shape type. Default is 0. The possible values for this attribute are defined by the W3C XML Schema float datatype.

strokecolor (Shape Stroke Color)	Specifies the primary color of the brush to use to stroke the path of the shape. Default is `black`. The color attribute of the stroke element (§19.1.2.21) overrides this. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*:
	
	`<v:shape … strokecolor="red" …>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

stroked (Shape Stroke Toggle)	Specifies whether the path defining the shape is stroked with a solid line. The stroke element (§19.1.2.21) defines other strokes. The on attribute of the stroke element overrides this attribute. Default is `true`.
	
	[*Example*:
	
	`<v:shape … fillcolor="red"`
	`  stroked="false" strokecolor="blue"…>`

Attributes	Description
	`</v:shape>`  *end example*] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

strokeweight (Shape Stroke Weight)	Specifies the width of the brush to use to stroke the path. Default is 1 point. If a number is given without units, the emu is used. The weight attribute of the stroke element (§19.1.2.21) overrides this attribute.
	
	[*Example*:
	
	`<v:shape … strokeweight="3pt" … >`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

style (Shape Styling Properties)	Specifies the CSS2 styling properties of the shape. The CSS2 (Cascading Style Sheets, Level 2) specification, a Recommendation of the World Wide Web Consortium, is available here: http://www.w3.org/TR/REC-CSS2.
	
	This attribute uses a semi-colon delimited list of “name:value” pairs, the syntax defined by section 4.1.8 of the CSS2 specification for grouped declarations (without the surrounding braces). The following tables define the allowed properties and the VML treatment of each.
	
	[*Example*:
	`<v:shape … style=’position:absolute;width:100pt;height:50pt’ … >`
	`</v:shape>`
	end example]
	
	Property — **Description**
	`flip` — Specifies that the orientation of a shape is flipped. Default is no value. Allowed values are:

Attributes	Description	
		• x - Flip along the y-axis, reversing the x-coordinates. • y - Flip along the x-axis, reversing the y-coordinates. • xy - Flip along both the y- and x-axis. • yx - Flip along both the x- and y-axis.
	`height`	Specifies the height of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`left`	Specifies the position of the left of the containing block of the shape relative to the element left of it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-bottom`	Specifies the position of the bottom of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`margin-left`	Specifies the position of the left of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of

Attributes	Description
	the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-right` — Specifies the position of the right of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-top` — Specifies the position of the top of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`mso-position-horizontal` — Specifies the horizontal positioning data for objects in WordprocessingML documents. Default is `absolute`. Allowed values are: • `absolute` • `left` • `center` • `right` • `inside` • `outside`

Attributes	Description	
	`mso-position-horizontal-relative`	Specifies relative horizontal position data for objects in WordprocessingML documents. This modifies the `mso-position-horizontal` property. Default is `text`. Allowed values are: • `margin` • `page` • `text` • `char`
	`mso-position-vertical`	Specifies the vertical positioning data for objects in WordprocessingML documents. Default is `absolute`. Allowed values are: • `absolute` • `top` • `center` • `bottom` • `inside` • `outside`
	`mso-position-vertical-relative`	Specifies relative vertical position data for objects in WordprocessingML documents. This modifies the `mso-position-vertical` property. Default is `text`. Allowed values are: • `margin` • `page` • `text` • `line`
	`mso-wrap-distance-bottom`	Specifies the distance from the bottom of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	`mso-wrap-distance-left`	Specifies the distance from the left side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	`mso-wrap-distance-right`	Specifies the distance from the right side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change

Attributes	Description
	the origin.
`mso-wrap-distance-top`	Specifies the distance from the top of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
`mso-wrap-edited`	Specifies whether the wrap coordinates were customized by the user. If the wrap coordinates are generated by an editor, this property is `true`; otherwise they were customized by a user. Default is `false`.
`mso-wrap-style`	Specifies the wrapping mode for text in shapes in WordprocessingML documents. Default is `square`. Allowed values are: • `square` - Wraps text inside the shape in a square. • `none` - Text does not wrap.
`position`	Specifies the kind of positioning used to place an element. Default is `static`. When the element is contained inside a group, this property shall be `absolute`. Allowed values are: • `static` - The element is positioned according to the normal flow of the page. The `top` and `left` properties are ignored. If the object is anchored inline, this value is used. • `absolute` - The element is positioned relative to the parent, using the `top` and `left` properties. • `relative` - The element is positioned according to the normal flow of the page, but the `top` and `left` properties are used. The overlap of overlapping elements is governed by the `z-index` property.
`rotation`	Specifies the angle that a shape is rotated, in degrees. Default is 0. Positive angles are clockwise.
`top`	Specifies the position of the top of the containing block of the shape relative to the element above it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed.

Attributes	Description
	<table><tr><td></td><td>• <percentage>- Value expressed as a percentage of the parent object's height.</td></tr><tr><td>`visibility`</td><td>Specifies whether a shape is displayed. Only `inherit` and `hidden` are used; any other values are mapped to `inherit`. Default is `inherit`. Allowed values are: • `hidden` - The shape is not visible, but is still part of the flow of the objects in the browser. Mouse events are not processed. • `inherit` - The visibility state is inherited from the parent of the shape.</td></tr><tr><td>`width`</td><td>Specifies the width of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.</td></tr><tr><td>`z-index`</td><td>Specifies the display order of overlapping shapes. Default is 0. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Uses the order that the shapes appear in the page, bottom to top. • <order>- A number that represents the stacking precedence. Shapes with higher numbers are placed on top of those with lower numbers. Negative numbers are allowed.</td></tr></table> The following properties are only used by the textbox element (§19.1.2.22): <table><tr><th>Property</th><th>Description</th></tr><tr><td>`direction`</td><td>Specifies the direction of the text in the textbox. Default is `ltr`. This property is superceded by the `mso-direction-alt` property if that is specified. Allowed values are: • `ltr` - Text is displayed left-to-right. • `rtl` - Text is displayed right-to-left.</td></tr></table>

Attributes	Description	
	`layout-flow`	Determines the flow of the text layout in a textbox. Default is `horizontal`. Allowed values are: • `horizontal` - Text is displayed horizontally. • `vertical` - Text is displayed vertically. • `vertical-ideographic` - Ideographic text is displayed vertically. • `horizontal-ideographic` - Ideographic text is displayed horizontally.
	`mso-direction-alt`	Specifies an alternate direction for text in textboxes. Overrides the `direction` property. The only allowed value is `context`.
	`mso-fit-shape-to-text`	Specifies whether the shape stretches to fit the text in the textbox. Default is `false`.
	`mso-fit-text-to-shape`	Specifies whether the text stretches to fit the textbox. Default is `false`.
	`mso-layout-flow-alt`	Specifies the alternate layout flow for text in textboxes. This property is used instead of `layout-flow` when the layout flow is from bottom to top for non-ideographic languages. Its only value is `bottom-to-top`.
	`mso-next-textbox`	Specifies the ID of the next textbox in a series. Used to keep track of a set of linked textboxes. Default is no value.
	`mso-rotate`	Specifies a specific rotation value for text in a textbox. Default is 0. Allowed values are: • 0 • 90 • 180 • -90
	`mso-text-scale`	Specifies the scaling factor for fitting text to shapes. Default is 0. This property is only used if `mso-fit-text-to-shape` is `true`.
	`v-text-anchor`	Specifies the vertical anchoring of text in a textbox. Default is `top`. The alignment of a text anchor only becomes evident if `mso-fit-text-to-shape` is `false`. This property is different from the `vertical-align` CSS property, which is used for ideographic languages. Allowed values are: • `top` • `middle`

Attributes	Description
	• bottom • top-center • middle-center • bottom-center • top-baseline • bottom-baseline • top-center-baseline • bottom-center-baseline

The following properties are only used by the textpath element (§19.1.2.23):

Property	Description
font	Specifies a compound value of font settings. Default is no value. The values are the same as those of the CSS font property. The order of definitions in the string is: font-style, font-variant, font-weight, font-size, line-height, font-family.
font-family	Specifies the family of the font. Default is no value. The values are the same as those of the CSS font-family property.
font-size	Specifies the size of the font. Default is no value. The font size is defined in points. The values are the same as those of the CSS font-size property.
font-style	Specifies the amount of slant for a font. Default is normal. The values are the same as those of the CSS font-style property. Allowed values are: • normal • italic • oblique - Treated the same as italic.
font-variant	Specifies the variant style of a font. Default is normal. The values are the same as those of the CSS font-variant property. Allowed values are: • normal • small-caps
font-weight	Specifies the thickness of the letters of the font. Default is normal. The values are the same as those of the CSS font-weight property. Allowed values are:

Value	Description
normal	Treated as non-bold.

Attributes	Description
	<table><tr><td></td><td><table><tr><td>lighter 100 200 300 400</td><td></td></tr><tr><td>bold bolder 500 600 700 800 900</td><td>Treated as bold.</td></tr></table></td></tr><tr><td>mso-text-shadow</td><td>Specifies whether a shadow is applied to the text on a text path. Default is false.</td></tr><tr><td>text-decoration</td><td>Specifies the style of text decoration. Default is none. The values are the same as those of the CSS text-decoration property. Allowed values are: • none • underline • overline • line-through • blink</td></tr><tr><td>v-rotate-letters</td><td>Specifies whether the letters of the text are rotated counterclockwise by 90 degrees. Default is false.</td></tr><tr><td>v-same-letter-heights</td><td>Specifies whether all letters are the same height regardless of initial case. If true, the lowercase letters are stretched to the height of the uppercase letters. Default is false.</td></tr><tr><td>v-text-align</td><td>Specifies the alignment of text. Default is left. Allowed values are: • left • right • center</td></tr></table>

Attributes	Description

	• justify • letter-justify - Distributes the extra space between the letters. • stretch-justify - Stretches the letters to fill in the space.
v-text-kern	Specifies whether kerning is turned on. Default is false.
v-text-reverse	Specifies whether the layout order of rows is reversed. Default is false. This is used for vertical text layout.
v-text-spacing-mode	Specifies the mode for letter spacing. Default is tightening. This property determines whether space is removed between each letter (tightening) or added between each letter (tracking). The amount of letter spacing change is defined by the v-text-spacing property. Allowed values are: • tightening • tracking
v-text-spacing	Specifies the amount of spacing for text in 100ths of single line spacing. Default is 100.

The line (§19.1.2.12), polyline (§19.1.2.15) and curve (§19.1.2.3) elements ignore the following properties:

- top
- left
- width
- height

The following properties are not inherited by an element that references a shapetype element (§19.1.2.20) via the id attribute:

- flip
- height
- left
- margin-left
- margin-top
- position
- rotation
- top
- visibility
- width
- z-index

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>The possible values for this attribute are defined by the W3C XML Schema string datatype.</td></tr>
<tr><td>target (Hyperlink Display Target)</td><td>Specifies a frame or window that a URL is displayed in. Default is no value. Allowed values are:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><targetname></td><td>String containing the name of the frame or window in which to load the document.</td></tr>
<tr><td>_blank</td><td>Specifies that the linked document is loaded into a new blank window. This window is not named.</td></tr>
<tr><td>_media</td><td>Specifies that the linked document is loaded into the browser’s multimedia pane.</td></tr>
<tr><td>_parent</td><td>Specifies that the linked document is loaded into the immediate parent of the document containing the link.</td></tr>
<tr><td>_search</td><td>Specifies that the linked document is loaded into the browser's search pane.</td></tr>
<tr><td>_self</td><td>Specifies that the linked document is loaded into the window in which the link was clicked (the active window).</td></tr>
<tr><td>_top</td><td>Specifies that the linked document is loaded into the topmost window.</td></tr>
</table>
[Example:
<pre>
<v:shape …
  href="http://www.openxmlformats.org"
  target="_self" … >
</v:shape>
</pre>
end example]
The possible values for this attribute are defined by the W3C XML Schema string datatype.</td></tr>
<tr><td>title (Shape Title)</td><td>Specifies the text displayed when the mouse pointer moves over the shape. Default is no value.
[Example:
<pre>
<v:shape … title="tooltip" … >
</v:shape>
</pre>
end example]
The possible values for this attribute are defined by the W3C XML Schema string</td></tr>
</table>

Attributes	Description
	datatype.

userdrawn (Exists In Master Slide)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the user has added the shape to a master slide. Default is `false`. Used by PresentationML.
	
	[*Example*:
	
	`<v:shape … o:userdrawn="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

userhidden (Hide Script Anchors)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a script anchor is hidden. Default is `false`. If `true`, script anchors stay hidden even if the shape is otherwise visible. A script anchor is the visual representation of a script that when displayed in an application.
	
	[*Example*:
	
	`<v:shape … o:userhidden="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

wrapcoords (Shape Bounding Polygon)	Specifies the bounding polygon that surrounds a shape. This is specified using a comma-delimited list of x and y coordinates: "x1,y1,x2,y2,x3,y3,…" This is used when text is tightly wrapped around a shape. Default is no value until the mso-wrap-mode style attribute is set to `tight` or `through`.
	
	[*Example*:
	
	`<v:shape …`
	`wrapcoords="0,0 0,200, 200,200, 200,0" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

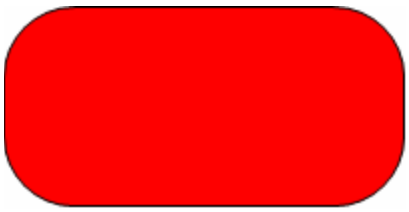
[*Note*: The W3C XML Schema definition of this element's content model (CT_Rect) is located in §A.6.1. *end note*]

19.1.2.17 roundrect (Rounded Rectangle)

This element is used to draw a rectangle with rounded corners. The CSS2 style content width and height define the width and height of the rectangle.

[*Example*:

```
<v:roundrect fillcolor="red" arcsize="35%"
  style="position:relative;top:0;left:0;width:200;height:100">
</v:roundrect>
```



end example]

Attributes	Description

allowincell (Allow in Table Cell)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can be placed in a table. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:allowincell="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

allowoverlap (Allow Shape Overlap)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can overlap another shape. Default is `true`. If `false`, the shape shifts left or right so as not to overlap another shape, similar to the behavior of the HTML float attribute.
	
	[*Example*:
	
	`<v:shape … o:allowoverlap="false" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

alt (Alternate Text)	Specifies alternative text describing the graphical object. This text should provide a brief description of the shape for use by accessibility tools. Default is no value.

Attributes	Description

	[*Example*: The alt text describes the basic shape:
	
	`<v:shape … fillcolor="red"`
	`  alt="Red rectangle">`
	`</v:shape>`
	
	The alt text describes the contents of a shape displaying an image:
	
	`<v:shape … alt="Picture of a sunset">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

arcsize (Rounded Corner Arc Size)	Specifies the amount of roundness for a rounded rectangle as a percentage of half the smaller dimension of the length and width of the rectangle. Default is 20%. An arc size of 0% yields square corners and 100% forms circular corners. A square with an arc size value of 100% is a circle. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. [*Example*: For example, a value of "52429f" represents 52429/65536 or 0.8. *end example*]
	
	[*Example*:
	
	`<v:roundrect … arcsize="35%">`
	`</v:roundrect>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderbottomcolor (Bottom Border Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the bottom border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderbottomcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderleftcolor (Border Left Color)	Specifies the left border color of an inline shape. Default is no value. [*Example*:

Attributes	Description

Namespace: urn:schemas-microsoft-com:office:office	`<v:shape … o:borderleftcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderrightcolor (Border Right Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the right border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderrightcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

bordertopcolor (Border Top Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the top border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:bordertopcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

bullet (Graphical Bullet)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the shape is a graphical bullet. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:bullet="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

button (Button Behavior Toggle) Namespace:	Specifies whether a shape exhibits button press behavior on click. Default is `false`. [*Example*:

Attributes	Description

urn:schemas-microsoft-com:office:office	`<v:shape … o:button="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

bwmode (Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies how a shape renders for black-and-white output devices. When a shape is printed on a black-and-white printer or displayed in a black-and-white view in an application, several options are possible. Default is auto, which uses o:bwnormal for normal black-and-white rendering and o:bwpure for pure black-and-white rendering.
	
	bwnormal and bwpure are subordinate to bwmode. If bwmode is "auto" then the value for bwnormal or bwpure is used depending on what the output format is. An application can define for itself what, if any, difference there is between normal B&W and pure B&W. [*Example*: Normal B&W might allow greyscale and pure B&W might not. *end example*]
	[*Example*: This shape renders in grayscale in a black-and-white environment:
	
	`<v:shape … o:bwmode="grayscale" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

bwnormal (Normal Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the black-and-white mode for normal black-and-white output devices. Default is auto.
	
	[*Example*: This shape renders in a pale grayscale in a normal black-and-white environment:
	
	`<v:shape …  o:bwmode="auto" o:bwnormal="lightgrayscale" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

bwpure (Pure Black-and-White Mode) Namespace: urn:schemas-	Specifies the black-and-white mode for pure black-and-white output devices. Default is auto. [*Example*: This shape renders in high contrast when in a pure black-and-white environment:

Attributes	Description

microsoft-com:office:office	`<v:shape … o:bwmode="auto" o:bwpure="highcontrast" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

chromakey (Image Transparency Color)	Specifies a color value that is transparent and show anything behind the shape. Default is no value.
	
	[*Example*:
	
	`<v:image … chromakey="white" …>`
	`</v:image>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

class (CSS Reference)	Specifies a reference to the definition of a CSS style. Default is no value.
	
	[*Example*: The snippets below are equivalent:
	
	`…`
	`.narrowstyle {width:50;height:100}`
	`…`
	`<v:shape … class="narrowstyle"`
	`  style="top:1;left:1">`
	`</v:shape>`
	
	`<v:shape … style="top:1;left:1;`
	`  width:50;height:100">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

clip (Clipping Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the clipping region is active. This is used in conjunction with the clippath (§19.2.2.3) element to create a clipping region.
	
	[*Example*:
	
	`<v:shape … o:clip="true">`
	`</v:shape>`

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

cliptowrap (Clip to Wrapping Polygon)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the clipping region for the shape aligns with the wrapping polygon that tightly surrounds the entire shape (essentially, that the shape shall not be drawn beyond its wrapping polygon's extents – if it does, the shape shall be clipped). Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:cliptowrap="true">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

connectortype (Shape Connector Type)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of connector used for joining shapes. Default is `straight`.
	
	[*Example*:
	
	`<v:shape … o:connectortype="elbow" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ConnectorType simple type (§19.2.3.7).

coordorigin (Coordinate Space Origin)	Specifies the coordinate of the top left corner of the shape's coordinate space. This determines the position of the (0,0) coordinate space origin within the shape's bounding box. Default is "0,0", which places the (0,0) origin at the top left corner of the bounding box.
	
	This affects shape properties that specify coordinate positions, such as the path attribute. Thus a path can be defined against a generic (0,0) origin and the coordorigin value translates the entire path within the shape's bounding space.
	
	[*Example*: The horizontal and vertical coordinate space ranges from -100 to +100 because the coordinate space (coordsize) is 200 by 200 and the top left coordinate is (-100,-100). The (0,0) origin lies at the center of the shape's bounding box, as evidenced by the position of the shape's path within the coordinate space:
	
	`<v:shape … coordsize="200,200"`
	`  coordorigin="-100,-100"`

Attributes	Description

	`path="m 0,0 l 0,50, 50,50, 50,0 x e">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

coordsize (Coordinate Space Size)	Specifies the size of the shape's coordinate space in coordinate units. Default is "1000,1000".
	
	The physical size of a coordinate unit length is determined by both the size of the coordinate space (coordsize) and the size of the shape (style width and height). The coordsize attribute defines the number of horizontal and vertical subdivisions into which the shape's bounding box is divided. The combination of coordsize and style width/height effective scales the shape anisotropically.
	
	[*Example*: The path is 50 units wide and tall, which is 25% of the size of the coordinate space:
	
	`<v:shape … coordsize="200,200"`
	`coordorigin="-100,-100"`
	`path="m 0,0 l 0,50, 50,50, 50,0 x e">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

dgmlayout (Diagram Node Layout Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of automatic layout to apply to the diagram node. This is only meaningful if the shape is a node in an organization chart, which is denoted by the `orgchart` value of the editas attribute of the group element.
	[*Example*:
	
	`<v:shape … dgmlayout="1">`
	`</v:shape>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).

dgmlayoutmru (Diagram Node Recent Layout Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of automatic layout most recently used on the child elements of the diagram node. This is only meaningful if the shape is a node in anorganization chart, which is denoted by the `orgchart` value of the editas attribute of the group element.
	
	[*Example*:
	
	`<v:shape … dgmlayout="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).

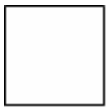
dgmnodekind (Diagram Node Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies an optional, application-defined parameter that is intended to be used by the application to tag different types of nodes in a diagram.
	
	[*Example*:
	
	`<v:shape … dgmnodekind="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

doubleclicknotify (Double-click Notification Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that an event message is sent when a shape is double-clicked. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:doubleclicknotify="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

fillcolor (Fill Color)	Specifies the color to use for the fill. Default is `white`. If the fill element (§19.1.2.5) is present, its color attribute takes precedence. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.

Attributes	Description

	[*Example*: This shape is red if its fill is visible:
	
	`<v:shape … fillcolor="red" … >`
	`</v:shape>`
	
	This is equivalent to:
	
	`<v:shape … fillcolor="#ff0000" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

filled (Shape Fill Toggle)	Specifies whether the closed path is filled. Default is `true`. This attribute is overridden by the fill on attribute.
	
	[*Example*:
	
	`<v:shape … filled="f"`
	`  fillcolor="red" …>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

forcedash (Force Dashed Outline)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a dashed outline is used to draw a shape when a shape has no line or fill. Default is `false`.
	
	Used by PresentationML placeholders to draw a dashed outline when there is no line and no fill for a shape.
	
	[*Example*:
	
	`<v:shape … o:forcedash="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

Attributes	Description

hr (Horizontal Rule Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that a shape is a horizontal rule. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hr="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hralign (Horizontal Rule Alignment)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the alignment of a horizontal rule. Default is `left`.
	
	[*Example*:
	
	`<v:shape … o:hralign="center" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_HrAlign simple type (§19.2.3.16).

href (Hyperlink Target)	Specifies a hyperlink URL target for the shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … href="http://www.openxmlformats.org" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

hrnoshade (Horizontal Rule 3D Shading Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the horizontal rule does not have 3-D shading. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hrnoshade="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hrpct (Horizontal	Specifies the length of a horizontal rule as a percentage of page width. Default is 0.

Attributes	Description

Rule Length Percentage)
 	
Namespace: urn:schemas-microsoft-com:office:office
	[*Example*:
	
	`<v:shape … o:hrpct="85" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema float datatype.

hrstd (Horizontal Rule Standard Display Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape is displayed as a standard horizontal rule. Only applies if hr is `true`. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hrstd="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

id (Unique Identifier)	Specifies a unique identifier that can be used to reference a VML object.
	
	Default is no value.
	
	[*Example*:
	
	`<v:shape … id="myShape" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

insetmode (Text Inset Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the application calculates the internal text margin instead of using the inset attribute. Default is `custom`. This attribute is only meaningful for text boxes.
	
	[*Example*:
	
	`<v:shape … o:insetmode="auto" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_InsetMode simple type

Attributes	Description
	(§19.2.3.17).

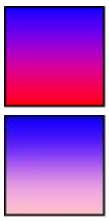
insetpen (Inset Border From Path)	Specifies that the border shall be displayed inside of the path defining the shape, rather than along the path (the default border placement), or outside of the path as might be done with an image.
	
	[*Example*:
	
	`<v:shape … insetpen="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

ole (Embedded Object Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the shape is an embedded object. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:ole="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

oleicon (Embedded Object Icon Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether an embedded object is displayed as an icon. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oleicon="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

oned (Shape Handle Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the extra handles of a shape are hidden. If `true`, hides all shape handles except the top left and bottom right; that is, the same handles that are used for a straight line segment. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oned="true" … >`
	`</v:shape>`

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

opacity (Fill Color Opacity)	Specifies the opacity of the primary fill color. Default is 1.0. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. [*Example*: A value of "52429f" represents 52429/65536 or 0.8. *end example*]
	
	[*Example*: The red color is 25% opaque:
	
	`<v:fill type="gradient" color="red"`
	`  color2="blue" opacity=".25">`
	`</v:fill>`
	
	 
	`opacity="1"`
	
	`opacity=".25"`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

preferrelative (Relative Resize Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the original size of an object is saved after reformatting. Default is `false`. If `true`, the original size of the object is stored and all resizing is based on a percentage of that original size. Otherwise, each resizing resets the scale to 100%.
	
	[*Example*:
	
	`<v:shape … o:preferrelative="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

print (Print Toggle)	Specifies whether the shape is printed. Default is `true`.
	
	[*Example*:
	
	`<v:shape … print="false" … >`
	`</v:shape>`

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

regroupid (Regroup ID)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies a previous group for a shape. An ID number is used to identify groups of shapes that are no longer grouped. This allows shapes to be regrouped programmatically.
	
	[*Example*: The shape was part of a group identified by the ID 040754:
	
	`<v:shape … o:regroupid="040754" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

spid (Optional String) Namespace: urn:schemas-microsoft-com:office:office	Specifies an optional string that an application can use to Identify the particular shape. Default is no value. The possible values for this attribute are defined by the W3C XML Schema string datatype.
spt (Optional Number) Namespace: urn:schemas-microsoft-com:office:office	Specifies an optional number that an application can use to associate the particular shape with a defined shape type. Default is 0. The possible values for this attribute are defined by the W3C XML Schema float datatype.

strokecolor (Shape Stroke Color)	Specifies the primary color of the brush to use to stroke the path of the shape. Default is `black`. The color attribute of the stroke element (§19.1.2.21) overrides this. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*:
	
	`<v:shape … strokecolor="red" …>`
	`</v:shape>`
	
	 

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

stroked (Shape Stroke Toggle)	Specifies whether the path defining the shape is stroked with a solid line. The stroke element (§19.1.2.21) defines other strokes. The on attribute of the stroke element overrides this attribute. Default is true.
	
	[*Example*:
	
	`<v:shape … fillcolor="red"`
	`  stroked="false" strokecolor="blue"…>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

strokeweight (Shape Stroke Weight)	Specifies the width of the brush to use to stroke the path. Default is 1 point. If a number is given without units, the emu is used. The weight attribute of the stroke element (§19.1.2.21) overrides this attribute.
	
	[*Example*:
	
	`<v:shape … strokeweight="3pt" … >`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

style (Shape Styling Properties)	Specifies the CSS2 styling properties of the shape. The CSS2 (Cascading Style Sheets, Level 2) specification, a Recommendation of the World Wide Web Consortium, is available here: http://www.w3.org/TR/REC-CSS2. This attribute uses a semi-colon delimited list of “name:value” pairs, the syntax defined by section 4.1.8 of the CSS2 specification for grouped declarations (without the

Attributes	Description

	surrounding braces). The following tables define the allowed properties and the VML treatment of each.
	
	[*Example*:
	`<v:shape … style='position:absolute;width:100pt;height:50pt' …`
	`>`
	`</v:shape>`
	end example]

Property	Description
`flip`	Specifies that the orientation of a shape is flipped. Default is no value. Allowed values are: • x - Flip along the y-axis, reversing the x-coordinates. • y - Flip along the x-axis, reversing the y-coordinates. • xy - Flip along both the y- and x-axis. • yx - Flip along both the x- and y-axis.
`height`	Specifies the height of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
`left`	Specifies the position of the left of the containing block of the shape relative to the element left of it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
`margin-bottom`	Specifies the position of the bottom of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of

Attributes	Description	
		the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`margin-left`	Specifies the position of the left of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-right`	Specifies the position of the right of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-top`	Specifies the position of the top of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the

Attributes	Description
	parent object's height.
`mso-position-horizontal`	Specifies the horizontal positioning data for objects in WordprocessingML documents. Default is `absolute`. Allowed values are: • `absolute` • `left` • `center` • `right` • `inside` • `outside`
`mso-position-horizontal-relative`	Specifies relative horizontal position data for objects in WordprocessingML documents. This modifies the `mso-position-horizontal` property. Default is `text`. Allowed values are: • `margin` • `page` • `text` • `char`
`mso-position-vertical`	Specifies the vertical positioning data for objects in WordprocessingML documents. Default is `absolute`. Allowed values are: • `absolute` • `top` • `center` • `bottom` • `inside` • `outside`
`mso-position-vertical-relative`	Specifies relative vertical position data for objects in WordprocessingML documents. This modifies the `mso-position-vertical` property. Default is `text`. Allowed values are: • `margin` • `page` • `text` • `line`
`mso-wrap-distance-`	Specifies the distance from the bottom of the shape to the text that wraps around it. Default is 0 pt. This property is different

Attributes	Description
	`bottom` — from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	`mso-wrap-distance-left` — Specifies the distance from the left side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	`mso-wrap-distance-right` — Specifies the distance from the right side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	`mso-wrap-distance-top` — Specifies the distance from the top of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	`mso-wrap-edited` — Specifies whether the wrap coordinates were customized by the user. If the wrap coordinates are generated by an editor, this property is `true`; otherwise they were customized by a user. Default is `false`.
	`mso-wrap-style` — Specifies the wrapping mode for text in shapes in WordprocessingML documents. Default is `square`. Allowed values are: • `square` - Wraps text inside the shape in a square. • `none` - Text does not wrap.
	`position` — Specifies the kind of positioning used to place an element. Default is `static`. When the element is contained inside a group, this property shall be `absolute`. Allowed values are: • `static` - The element is positioned according to the normal flow of the page. The `top` and `left` properties are ignored. If the object is anchored inline, this value is used. • `absolute` - The element is positioned relative to the parent, using the `top` and `left` properties. • `relative` - The element is positioned according to the normal flow of the page, but the `top` and `left` properties are used. The overlap of overlapping elements is governed by the `z-index` property.

Attributes	Description	
	`rotation`	Specifies the angle that a shape is rotated, in degrees. Default is 0. Positive angles are clockwise.
	`top`	Specifies the position of the top of the containing block of the shape relative to the element above it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`visibility`	Specifies whether a shape is displayed. Only `inherit` and `hidden` are used; any other values are mapped to `inherit`. Default is `inherit`. Allowed values are: • `hidden` - The shape is not visible, but is still part of the flow of the objects in the browser. Mouse events are not processed. • `inherit` - The visibility state is inherited from the parent of the shape.
	`width`	Specifies the width of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`z-index`	Specifies the display order of overlapping shapes. Default is 0. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Uses the order that the shapes appear in the page, bottom to top. • <order>- A number that represents the stacking precedence. Shapes with higher numbers are placed on

Attributes	Description
	top of those with lower numbers. Negative numbers are allowed.

The following properties are only used by the textbox element (§19.1.2.22):

Property	Description
`direction`	Specifies the direction of the text in the textbox. Default is `ltr`. This property is superceded by the `mso-direction-alt` property if that is specified. Allowed values are: • `ltr` - Text is displayed left-to-right. • `rtl` - Text is displayed right-to-left.
`layout-flow`	Determines the flow of the text layout in a textbox. Default is `horizontal`. Allowed values are: • `horizontal` - Text is displayed horizontally. • `vertical` - Text is displayed vertically. • `vertical-ideographic` - Ideographic text is displayed vertically. • `horizontal-ideographic` - Ideographic text is displayed horizontally.
`mso-direction-alt`	Specifies an alternate direction for text in textboxes. Overrides the `direction` property. The only allowed value is `context`.
`mso-fit-shape-to-text`	Specifies whether the shape stretches to fit the text in the textbox. Default is `false`.
`mso-fit-text-to-shape`	Specifies whether the text stretches to fit the textbox. Default is `false`.
`mso-layout-flow-alt`	Specifies the alternate layout flow for text in textboxes. This property is used instead of `layout-flow` when the layout flow is from bottom to top for non-ideographic languages. Its only value is `bottom-to-top`.
`mso-next-textbox`	Specifies the ID of the next textbox in a series. Used to keep track of a set of linked textboxes. Default is no value.
`mso-rotate`	Specifies a specific rotation value for text in a textbox. Default is 0. Allowed values are: • 0

Attributes	Description

	• 90 • 180 • -90
`mso-text-scale`	Specifies the scaling factor for fitting text to shapes. Default is 0. This property is only used if `mso-fit-text-to-shape` is `true`.
`v-text-anchor`	Specifies the vertical anchoring of text in a textbox. Default is `top`. The alignment of a text anchor only becomes evident if `mso-fit-text-to-shape` is `false`. This property is different from the `vertical-align` CSS property, which is used for ideographic languages. Allowed values are: • `top` • `middle` • `bottom` • `top-center` • `middle-center` • `bottom-center` • `top-baseline` • `bottom-baseline` • `top-center-baseline` • `bottom-center-baseline`

The following properties are only used by the textpath element (§19.1.2.23):

Property	Description
`font`	Specifies a compound value of font settings. Default is no value. The values are the same as those of the CSS font property. The order of definitions in the string is: font-style, font-variant, font-weight, font-size, line-height, font-family.
`font-family`	Specifies the family of the font. Default is no value. The values are the same as those of the CSS font-family property.
`font-size`	Specifies the size of the font. Default is no value. The font size is defined in points. The values are the same as those of the CSS font-size property.
`font-style`	Specifies the amount of slant for a font. Default is `normal`. The values are the same as those of the CSS font-style property. Allowed values are: • `normal` • `italic` • `oblique` - Treated the same as italic.

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>font-variant</td><td>Specifies the variant style of a font. Default is normal. The values are the same as those of the CSS font-variant property. Allowed values are:

• normal
• small-caps</td></tr>
<tr><td>font-weight</td><td>Specifies the thickness of the letters of the font. Default is normal. The values are the same as those of the CSS font-weight property. Allowed values are:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>normal

lighter

100

200

300

400</td><td>Treated as non-bold.</td></tr>
<tr><td>bold

bolder

500

600

700

800

900</td><td>Treated as bold.</td></tr>
</table></td></tr>
<tr><td>mso-text-shadow</td><td>Specifies whether a shadow is applied to the text on a text path. Default is false.</td></tr>
<tr><td>text-decoration</td><td>Specifies the style of text decoration. Default is none. The values are the same as those of the CSS text-decoration property. Allowed values are:

• none
• underline
• overline
• line-through</td></tr>
</table>
</td></tr>
</table>

Attributes	Description

	• blink
v-rotate-letters	Specifies whether the letters of the text are rotated counterclockwise by 90 degrees. Default is false.
v-same-letter-heights	Specifies whether all letters are the same height regardless of initial case. If true, the lowercase letters are stretched to the height of the uppercase letters. Default is false.
v-text-align	Specifies the alignment of text. Default is left. Allowed values are: • left • right • center • justify • letter-justify - Distributes the extra space between the letters. • stretch-justify - Stretches the letters to fill in the space.
v-text-kern	Specifies whether kerning is turned on. Default is false.
v-text-reverse	Specifies whether the layout order of rows is reversed. Default is false. This is used for vertical text layout.
v-text-spacing-mode	Specifies the mode for letter spacing. Default is tightening. This property determines whether space is removed between each letter (tightening) or added between each letter (tracking). The amount of letter spacing change is defined by the v-text-spacing property. Allowed values are: • tightening • tracking
v-text-spacing	Specifies the amount of spacing for text in 100ths of single line spacing. Default is 100.

The line (§19.1.2.12), polyline (§19.1.2.15) and curve (§19.1.2.3) elements ignore the following properties:

- top
- left
- width
- height

The following properties are not inherited by an element that references a shapetype element (§19.1.2.20) via the id attribute:

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>

flip
height
left
margin-left
margin-top
position
rotation
top
visibility
width
z-index

The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>target (Hyperlink Display Target)</td><td>
Specifies a frame or window that a URL is displayed in. Default is no value. Allowed values are:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><targetname></td><td>String containing the name of the frame or window in which to load the document.</td></tr>
<tr><td>_blank</td><td>Specifies that the linked document is loaded into a new blank window. This window is not named.</td></tr>
<tr><td>_media</td><td>Specifies that the linked document is loaded into the browser’s multimedia pane.</td></tr>
<tr><td>_parent</td><td>Specifies that the linked document is loaded into the immediate parent of the document containing the link.</td></tr>
<tr><td>_search</td><td>Specifies that the linked document is loaded into the browser's search pane.</td></tr>
<tr><td>_self</td><td>Specifies that the linked document is loaded into the window in which the link was clicked (the active window).</td></tr>
<tr><td>_top</td><td>Specifies that the linked document is loaded into the topmost window.</td></tr>
</table>
[Example:
<pre>
<v:shape …
  href="http://www.openxmlformats.org"
  target="_self" … >
</v:shape>
</pre>
end example]
</td></tr>
</table>

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

title (Shape Title)	Specifies the text displayed when the mouse pointer moves over the shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … title="tooltip" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

userdrawn (Exists In Master Slide)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the user has added the shape to a master slide. Default is `false`. Used by PresentationML.
	
	[*Example*:
	
	`<v:shape … o:userdrawn="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

userhidden (Hide Script Anchors)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a script anchor is hidden. Default is `false`. If `true`, script anchors stay hidden even if the shape is otherwise visible. A script anchor is the visual representation of a script that when displayed in an application.
	
	[*Example*:
	
	`<v:shape … o:userhidden="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

wrapcoords (Shape Bounding Polygon)	Specifies the bounding polygon that surrounds a shape. This is specified using a comma-delimited list of x and y coordinates: "x1,y1,x2,y2,x3,y3,…" This is used when text is tightly wrapped around a shape. Default is no value until the mso-wrap-mode style attribute is set to `tight` or `through`.

Attributes	Description

	[*Example*:
	
	`<v:shape …`
	`  wrapcoords="0,0 0,200, 200,200, 200,0" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_RoundRect) is located in §A.6.1. *end note*]

19.1.2.18 shadow (Shadow Effect)

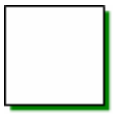
This element adds shadow effects to a shape. The `on` attribute shall be `true` for the shadow to be displayed.

[*Example*:

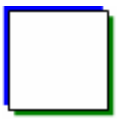
```
<v:shadow on="true" type="perspective"
  matrix="1.25,-2,,1.5,,.000001"
  offset="38pt,-6pt">
</v:shadow>
```

end example]

Attributes	Description

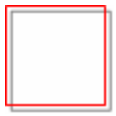
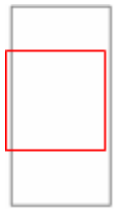
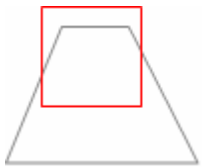
color (Shadow Primary Color)	Specifies the color of the primary shadow. Default is gray (RGB 128,128,128).
	
	[*Example*:
	
	`<v:shadow on="true" color="green">`
	`</v:shadow>`
	
	Applied to a simple square the shadow looks like this:
	
	 

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

color2 (Shadow Secondary Color)	Specifies the color of the second shadow, or highlight in an embossed or engraved shadow. Default is light gray (RGB 203,203,203).
	
	[*Example*:
	
	`<v:shadow on="true" type="double"`
	`  color="green" color2="blue">`
	`</v:shadow>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

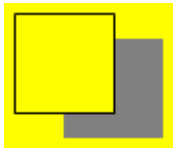
id (Unique Identifier)	Specifies a unique identifier that can be used to reference a VML object.
	
	Default is no value.
	
	[*Example*:
	
	`<v:shape … id="myShape" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

matrix (Shadow Perspective Matrix)	Specifies a perspective transform for a shadow. Default is no value. The matrix is given in the form "s_{xx}, s_{xy}, s_{yx}, s_{yy}, p_x, p_y" where s = scale and p = perspective. If the offset attribute is in absolute units then p_x, p_y are in 1/EMU units; otherwise they are an inverse fraction of the shape size. [*Example*: The following snippets explain the matrix parameters. The shadow is applied to a simple square with no fill and a red stroke color (note there is a default shadow offset):

Attributes	Description
	 `matrix=",,,,,"` s_{xx}, s_{yy} specify scaling factors for the x and y dimensions:  `matrix="2,,,,,"`  `matrix=",,,2,,"` s_{xy}, s_{yx} specify skews in the x and y dimensions:  `matrix=",2,,,,"`  `matrix=",,-2,,,"` p_x, p_y effectively set the perspective trapezoid skews along the x and y dimensions:  `matrix=",,,,.000001,"`  `matrix=",,,,,-.000002"`

Attributes	Description

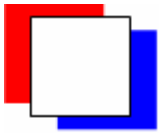
	end example]
	
	[*Example*:
	
	`<v:shadow on="true" type="perspective"`
	`  matrix="1.25,-2,,1.5,,.000001"`
	`  offset="38pt,-6pt">`
	`</v:shadow>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

obscured (Shadow Transparency)	Specifies whether a shadow is transparent. Default is `false`. If `true`, the shadow is transparent if there is no fill on the shape.
	
	[*Example*:
	
	`<v:background fillcolor="yellow"/>`
	`<v:shape style="width:50;height:50"`
	`  filled="false" fillcolor="red"`
	`  path="m 0,0 l 0,1000 1000,1000 1000,0 x e">`
	`  <v:shadow on="true" offset="50%,25%"`
	`  obscured="true">`
	`  </v:shadow>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

offset (Shadow Primary Offset)	Specifies the primary shadow's x,y offset from the shape's location. Default is "2pt,2pt". Values are either an absolute measurement or a fractional value of the shape dimensions, from –50% to 50%.

Attributes	Description

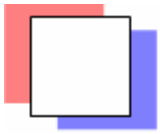
	[*Example*:
	
	`<v:shadow on="true" offset="50%,25%">`
	`</v:shadow>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

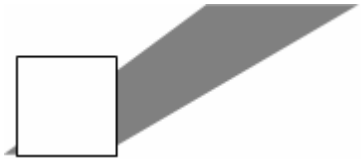
offset2 (Shadow Secondary Offset)	Specifies the secondary shadow's x,y offset from the shape's location. Default is "-2pt,-2pt".
	
	[*Example*:
	
	`<v:shadow type="double" on="true"`
	`color="blue" offset="10pt,5pt"`
	`color2="red" offset2="-10pt,-5pt">`
	`</v:shadow>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

on (Shadow Toggle)	Specifies whether to show a shadow. Default is `true`. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

opacity (Shadow Opacity)	Specifies the opacity of the shadow. Default is 1. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. [*Example*: For example, a value of "52429f" represents 52429/65536 or 0.8. *end example*]
	
	[*Example*:
	
	`<v:shadow type="double" on="true" opacity=".5"`
	`color="blue" offset="10pt,5pt"`

Attributes	Description

	`color2="red" offset2="-10pt,-5pt">`
	`</v:shadow>`
	 
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

origin (Shadow Origin)	Specifies the center of the shadow relative to the shape's origin. Specified as a pair of fractional values of the shape dimensions, ranging from 50% to -50%. Default is "0,0".
	
	[*Example*: This example is unchanged from above except for the addition of the origin attribute:
	
	`<v:shadow on="true" type="perspective"`
	`matrix="1.25,-2,,1.5,,.000001"`
	`offset="38pt,-6pt" origin="10%,-10%">`
	`</v:shadow>`
	 
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

type (Shadow Type)	Specifies the kind ofshadow. Default is `single`. Allowed values are: Value: Description `single`: Single shadow. `double`: Double shadow. color2 and offset2 are used for the second shadow's color and offset. `perspective`: Perspective shadow. `shaperelative`: The shadow is created relative to the shape. `drawingrelative`: The shadow is created relative to the drawing.

Attributes	Description
	`emboss` — The shadow has an embossed look. The possible values for this attribute are defined by the ST_ShadowType simple type (§19.1.3.6).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Shadow) is located in §A.6.1. *end note*]

19.1.2.19 shape (Shape Definition)

This element is used to describe a shape, the core object in VML. This element can appear by itself or within a group element (§19.1.2.7). If a shapetype element (§19.1.2.20) is referenced using the type attribute, any attributes specified in the shape override those found in the shapetype.

[*Example*:

```
<v:shape style="position:absolute;top:50;left:20;width:50;height:50"
  path="m 0,0 l 0,1000 1000,1000 1000,0 x e">
  <v:shadow on="true" type="perspective"
    matrix="1.25,-2,,1.5,,.000001" offset="38pt,-6pt"/>
</v:shape>

<v:shape style="position:absolute;top:50;left:20;width:50;height:50"
  fillcolor="yellow" path="m 0,0 l 0,1000 1000,1000 1000,0 x e">
  <v:extrusion on="true" lightposition="0,-2000,10000"/>
</v:shape>
```

end example]

Attributes	Description
adj (Adjustment Parameters)	Specifies a comma-delimited list of parameters, or adjustment values, used to define values for a parameterized formula. Values can be omitted. There can be up to 8 adjust values. Each value is referenced using # followed by a number corresponding to the zero-based index for that value in the list of adjustment values. [*Example*: For example, #2 references the second value in the adj list. *end example*] [*Example*: The following shape uses formulas to define a simple rectangle. The adj values

Attributes	Description

	are referenced by the eqn attribute of the f element (§19.1.2.4) and in turn referenced by the path element (§19.1.2.14).
	
	`<v:shape`
	`  coordorigin="0 0" coordsize="200 200"`
	`  style="position:relative;top:30;left:30;width:20;height:20"`
	`  adj="1, 1, 1, 200, 200, 200, 200, 1">`
	`  <v:path v="m @0,@1 l @2,@3, @4,@5, @6,@7 x e"/>`
	`  <v:formulas>`
	`  <v:f eqn="val #0"/>`
	`  <v:f eqn="val #1"/>`
	`  <v:f eqn="val #2"/>`
	`  <v:f eqn="val #3"/>`
	`  <v:f eqn="val #4"/>`
	`  <v:f eqn="val #5"/>`
	`  <v:f eqn="val #6"/>`
	`  <v:f eqn="val #7"/>`
	`  </v:formulas>`
	`</v:shape>`
	
	This is the equivalent of:
	
	`<v:shape`
	`  coordorigin="0 0" coordsize="200 200"`
	`  style="position:relative;top:30;left:30;width:20;height:20"`
	`  path="m 1,1 l 1,200, 200,200, 200,1 x e">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

allowincell (Allow in Table Cell)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can be placed in a table. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:allowincell="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

allowoverlap (Allow Shape Overlap)	Specifies whether a shape can overlap another shape. Default is `true`. If `false`, the shape shifts left or right so as not to overlap another shape, similar to the behavior of the HTML float attribute.

Attributes	Description

Namespace: urn:schemas-microsoft-com:office:office	[*Example*:
	
	`<v:shape … o:allowoverlap="false" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

alt (Alternate Text)	Specifies alternative text describing the graphical object. This text should provide a brief description of the shape for use by accessibility tools. Default is no value.
	
	[*Example*: The alt text describes the basic shape:
	
	`<v:shape … fillcolor="red"`
	`  alt="Red rectangle">`
	`</v:shape>`
	
	The alt text describes the contents of a shape displaying an image:
	
	`<v:shape … alt="Picture of a sunset">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderbottomcolor (Bottom Border Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the bottom border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderbottomcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderleftcolor (Border Left Color)
 	
Namespace: urn:schemas-microsoft-
	Specifies the left border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderleftcolor="red" … >`
	`</v:shape>`

Attributes	Description
com:office:office	*end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderrightcolor (Border Right Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the right border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderrightcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

bordertopcolor (Border Top Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the top border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:bordertopcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

bullet (Graphical Bullet)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the shape is a graphical bullet. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:bullet="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

button (Button Behavior Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape exhibits button press behavior on click. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:button="true" … >`
	`</v:shape>`

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

bwmode (Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies how a shape renders for black-and-white output devices. When a shape is printed on a black-and-white printer or displayed in a black-and-white view in an application, several options are possible. Default is auto, which uses o:bwnormal for normal black-and-white rendering and o:bwpure for pure black-and-white rendering.
	
	bwnormal and bwpure are subordinate to bwmode. If bwmode is "auto" then the value for bwnormal or bwpure is used depending on what the output format is. An application can define for itself what, if any, difference there is between normal B&W and pure B&W. [*Example*: Normal B&W might allow greyscale and pure B&W might not. *end example*]
	[*Example*: This shape renders in grayscale in a black-and-white environment:
	
	`<v:shape … o:bwmode="grayscale" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

bwnormal (Normal Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the black-and-white mode for normal black-and-white output devices. Default is auto.
	
	[*Example*: This shape renders in a pale grayscale in a normal black-and-white environment:
	
	`<v:shape …  o:bwmode="auto" o:bwnormal="lightgrayscale" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

bwpure (Pure Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the black-and-white mode for pure black-and-white output devices. Default is auto.
	
	[*Example*: This shape renders in high contrast when in a pure black-and-white environment:
	
	`<v:shape … o:bwmode="auto" o:bwpure="highcontrast" … >`
	`</v:shape>`

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

chromakey (Image Transparency Color)	Specifies a color value that is transparent and show anything behind the shape. Default is no value.
	
	[*Example*:
	
	`<v:image … chromakey="white" …>`
	`</v:image>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

class (CSS Reference)	Specifies a reference to the definition of a CSS style. Default is no value.
	
	[*Example*: The snippets below are equivalent:
	
	`…`
	`.narrowstyle {width:50;height:100}`
	`…`
	`<v:shape … class="narrowstyle"`
	`  style="top:1;left:1">`
	`</v:shape>`
	
	`<v:shape … style="top:1;left:1;`
	`  width:50;height:100">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

clip (Clipping Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the clipping region is active. This is used in conjunction with the clippath (§19.2.2.3) element to create a clipping region.
	
	[*Example*:
	
	`<v:shape … o:clip="true">`
	`</v:shape>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

cliptowrap (Clip to Wrapping Polygon)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the clipping region for the shape aligns with the wrapping polygon that tightly surrounds the entire shape (essentially, that the shape shall not be drawn beyond its wrapping polygon's extents – if it does, the shape shall be clipped). Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:cliptowrap="true">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

connectortype (Shape Connector Type)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of connector used for joining shapes. Default is `straight`.
	
	[*Example*:
	
	`<v:shape … o:connectortype="elbow" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ConnectorType simple type (§19.2.3.7).

coordorigin (Coordinate Space Origin)	Specifies the coordinate of the top left corner of the shape's coordinate space. This determines the position of the (0,0) coordinate space origin within the shape's bounding box. Default is "0,0", which places the (0,0) origin at the top left corner of the bounding box.
	
	This affects shape properties that specify coordinate positions, such as the path attribute. Thus a path can be defined against a generic (0,0) origin and the coordorigin value translates the entire path within the shape's bounding space.
	
	[*Example*: The horizontal and vertical coordinate space ranges from -100 to +100 because the coordinate space (coordsize) is 200 by 200 and the top left coordinate is (-100,-100). The (0,0) origin lies at the center of the shape's bounding box, as evidenced by the position of the shape's path within the coordinate space:
	
	`<v:shape … coordsize="200,200"`
	`coordorigin="-100,-100"`
	`path="m 0,0 l 0,50, 50,50, 50,0 x e">`
	`</v:shape>`

Attributes	Description
	❑ *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

coordsize (Coordinate Space Size)	Specifies the size of the shape's coordinate space in coordinate units. Default is "1000,1000".
	
	The physical size of a coordinate unit length is determined by both the size of the coordinate space (coordsize) and the size of the shape (style width and height). The coordsize attribute defines the number of horizontal and vertical subdivisions into which the shape's bounding box is divided. The combination of coordsize and style width/height effective scales the shape anisotropically.
	
	[*Example*: The path is 50 units wide and tall, which is 25% of the size of the coordinate space:
	
	`<v:shape … coordsize="200,200"`
	`coordorigin="-100,-100"`
	`path="m 0,0 l 0,50, 50,50, 50,0 x e">`
	`</v:shape>`
	
	❑
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

dgmlayout (Diagram Node Layout Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of automatic layout to apply to the diagram node. This is only meaningful if the shape is a node in an organization chart, which is denoted by the `orgchart` value of the editas attribute of the group element.
	[*Example*:
	
	`<v:shape … dgmlayout="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).

Attributes	Description

dgmlayoutmru (Diagram Node Recent Layout Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of automatic layout most recently used on the child elements of the diagram node. This is only meaningful if the shape is a node in anorganization chart, which is denoted by the `orgchart` value of the editas attribute of the group element.
	
	[*Example*:
	
	`<v:shape … dgmlayout="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).

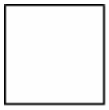
dgmnodekind (Diagram Node Identifier)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies an optional, application-defined parameter that is intended to be used by the application to tag different types of nodes in a diagram.
	
	[*Example*:
	
	`<v:shape … dgmnodekind="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

doubleclicknotify (Double-click Notification Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that an event message is sent when a shape is double-clicked. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:doubleclicknotify="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

equationxml (Storage for Alternate Math Content)	Specifies alternate XML markup which can be used to rehydrate an equation using the Office Open XML Math syntax. The actual format of the contents of this attribute is application-defined, but shall contain Office Open XML Math as well as any application-specific content. [*Note*: This form of storing alternate markup is inappropriate, and to be avoided in favor of the more flexible approach used by the child equationxml element (§19.2.2.10). *end note*] The XML markup stored in this attribute shall be escaped as needed to contain only those

Attributes	Description
	characters legal in an attribute value. The possible values for this attribute are defined by the W3C XML Schema string datatype.

fillcolor (Fill Color)	Specifies the color to use for the fill. Default is `white`. If the fill element (§19.1.2.5) is present, its color attribute takes precedence. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*: This shape is red if its fill is visible:
	
	`<v:shape … fillcolor="red" … >`
	`</v:shape>`
	
	This is equivalent to:
	
	`<v:shape … fillcolor="#ff0000" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

filled (Shape Fill Toggle)	Specifies whether the closed path is filled. Default is `true`. This attribute is overridden by the fill on attribute.
	
	[*Example*:
	
	`<v:shape … filled="f"`
	`  fillcolor="red" …>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

forcedash (Force Dashed Outline) Namespace:	Specifies whether a dashed outline is used to draw a shape when a shape has no line or fill. Default is `false`. Used by PresentationML placeholders to draw a dashed outline when there is no line and

Attributes	Description

urn:schemas-microsoft-com:office:office	no fill for a shape.
	
	[*Example*:
	
	`<v:shape … o:forcedash="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

gfxdata (Encoded Package)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies a base-64 encoded package as defined in ISO/IEC 29500-2 that contains DrawingML content as defined in ISO/IEC 29500-1. [*Rationale:* This attribute allows an application to use VML to represent graphical content for a legacy document while still persisting DrawingML for consuming applications that support DrawingML. For example, a diagram stored within this attribute would have the four parts defined for a DrawingML diagram, as well as any number of application-defined parts and relationships. *end rationale*]
	
	[*Example:* A DrawingML object is encoded in the gfxdata attribute, leaving VML to handle the visual display:
	
	`<v:shape id="Diagram 1" o:spid="_x0000_i1025"`
	`type="#_x0000_t75" style="width:446.25pt;height:252pt;`
	`visibility:visible" o:gfxdata="UEsDBBQABgAIAAAAIQDIu8KcTQE…">`
	`<v:imagedata r:id="rId4" o:title=""/>`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.

hr (Horizontal Rule Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that a shape is a horizontal rule. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hr="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hralign (Horizontal Rule Alignment)	Specifies the alignment of a horizontal rule. Default is `left`.

Attributes	Description

Namespace: urn:schemas-microsoft-com:office:office	[*Example*:
	
	`<v:shape … o:hralign="center" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_HrAlign simple type (§19.2.3.16).

href (Hyperlink Target)	Specifies a hyperlink URL target for the shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … href="http://www.openxmlformats.org" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

hrnoshade (Horizontal Rule 3D Shading Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the horizontal rule does not have 3-D shading. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hrnoshade="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hrpct (Horizontal Rule Length Percentage)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the length of a horizontal rule as a percentage of page width. Default is 0.
	
	[*Example*:
	
	`<v:shape … o:hrpct="85" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema float datatype.

hrstd (Horizontal Rule Standard Display Toggle)	Specifies whether a shape is displayed as a standard horizontal rule. Only applies if hr is `true`. Default is `false`. [*Example*:

Attributes	Description

Namespace: urn:schemas-microsoft-com:office:office	`<v:shape … o:hrstd="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

id (Unique Identifier)	Specifies a unique identifier that can be used to reference a VML object.
	
	Default is no value.
	
	[*Example*:
	
	`<v:shape … id="myShape" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

insetmode (Text Inset Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the application calculates the internal text margin instead of using the inset attribute. Default is `custom`. This attribute is only meaningful for text boxes.
	
	[*Example*:
	
	`<v:shape … o:insetmode="auto" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_InsetMode simple type (§19.2.3.17).

insetpen (Inset Border From Path)	Specifies that the border shall be displayed inside of the path defining the shape, rather than along the path (the default border placement), or outside of the path as might be done with an image.
	
	[*Example*:
	
	`<v:shape … insetpen="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type

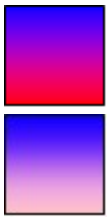
Attributes	Description
	(§20.1.2.5).

ole (Embedded Object Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the shape is an embedded object. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:ole="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

oleicon (Embedded Object Icon Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether an embedded object is displayed as an icon. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oleicon="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

oned (Shape Handle Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the extra handles of a shape are hidden. If `true`, hides all shape handles except the top left and bottom right; that is, the same handles that are used for a straight line segment. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oned="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

opacity (Fill Color Opacity)	Specifies the opacity of the primary fill color. Default is 1.0. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. [*Example*: A value of "52429f" represents 52429/65536 or 0.8. *end example*]
	
	[*Example*: The red color is 25% opaque:
	
	`<v:fill type="gradient" color="red"`
	`  color2="blue" opacity=".25">`
	`</v:fill>`

Attributes	Description
	 `opacity="1"` `opacity=".25"` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
path (Edge Path)	Specifies the line that makes up the edges of a shape. See the v attribute of the path element (§19.1.2.14) for a full description. The possible values for this attribute are defined by the W3C XML Schema string datatype.

preferrelative (Relative Resize Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the original size of an object is saved after reformatting. Default is `false`. If `true`, the original size of the object is stored and all resizing is based on a percentage of that original size. Otherwise, each resizing resets the scale to 100%.
	
	[*Example*:
	
	`<v:shape … o:preferrelative="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

print (Print Toggle)	Specifies whether the shape is printed. Default is `true`.
	
	[*Example*:
	
	`<v:shape … print="false" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

regroupid (Regroup ID) Namespace:	Specifies a previous group for a shape. An ID number is used to identify groups of shapes that are no longer grouped. This allows shapes to be regrouped programmatically. [*Example*: The shape was part of a group identified by the ID 040754:

Attributes	Description

urn:schemas-microsoft-com:office:office	`<v:shape … o:regroupid="040754" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

spid (Optional String) Namespace: urn:schemas-microsoft-com:office:office	Specifies an optional string that an application can use to Identify the particular shape. Default is no value. The possible values for this attribute are defined by the W3C XML Schema string datatype.
spt (Optional Number) Namespace: urn:schemas-microsoft-com:office:office	Specifies an optional number that an application can use to associate the particular shape with a defined shape type. Default is 0. The possible values for this attribute are defined by the W3C XML Schema float datatype.

strokecolor (Shape Stroke Color)	Specifies the primary color of the brush to use to stroke the path of the shape. Default is `black`. The color attribute of the stroke element (§19.1.2.21) overrides this. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*:
	
	`<v:shape … strokecolor="red" …>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

stroked (Shape Stroke Toggle)	Specifies whether the path defining the shape is stroked with a solid line. The stroke element (§19.1.2.21) defines other strokes. The on attribute of the stroke element overrides this attribute. Default is `true`.

Attributes	Description

	[*Example*:
	
	`<v:shape … fillcolor="red"`
	`  stroked="false" strokecolor="blue"…>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

strokeweight (Shape Stroke Weight)	Specifies the width of the brush to use to stroke the path. Default is 1 point. If a number is given without units, the emu is used. The weight attribute of the stroke element (§19.1.2.21) overrides this attribute.
	
	[*Example*:
	
	`<v:shape … strokeweight="3pt" … >`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

style (Shape Styling Properties)	Specifies the CSS2 styling properties of the shape. The CSS2 (Cascading Style Sheets, Level 2) specification, a Recommendation of the World Wide Web Consortium, is available here: http://www.w3.org/TR/REC-CSS2.
	
	This attribute uses a semi-colon delimited list of “name:value” pairs, the syntax defined by section 4.1.8 of the CSS2 specification for grouped declarations (without the surrounding braces). The following tables define the allowed properties and the VML treatment of each.
	
	[*Example*:
	`<v:shape … style=’position:absolute;width:100pt;height:50pt’ …`
	`>`
	`</v:shape>`
	end example]

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Property</th><th>Description</th></tr>
<tr><td>flip</td><td>Specifies that the orientation of a shape is flipped. Default is no value. Allowed values are:

• x - Flip along the y-axis, reversing the x-coordinates.
• y - Flip along the x-axis, reversing the y-coordinates.
• xy - Flip along both the y- and x-axis.
• yx - Flip along both the x- and y-axis.</td></tr>
<tr><td>height</td><td>Specifies the height of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are:

• auto - Default position of an element in the flow of the page.
• <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed.
• <percentage>- Value expressed as a percentage of the parent object's height.</td></tr>
<tr><td>left</td><td>Specifies the position of the left of the containing block of the shape relative to the element left of it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are:

• auto - Default position of an element in the flow of the page.
• <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed.
• <percentage>- Value expressed as a percentage of the parent object's width.</td></tr>
<tr><td>margin-bottom</td><td>Specifies the position of the bottom of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are:

• auto - Default position of an element in the flow of the page.
• <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed.
• <percentage>- Value expressed as a percentage of the parent object's height.</td></tr>
</table>
</td></tr>
</table>

Attributes	Description	
	`margin-left`	Specifies the position of the left of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-right`	Specifies the position of the right of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-top`	Specifies the position of the top of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`mso-position-horizontal`	Specifies the horizontal positioning data for objects in WordprocessingML documents. Default is `absolute`. Allowed values are: • `absolute` • `left` • `center` • `right`

Attributes	Description
	• inside • outside
mso-position-horizontal-relative	Specifies relative horizontal position data for objects in WordprocessingML documents. This modifies the mso-position-horizontal property. Default is text. Allowed values are: • margin • page • text • char
mso-position-vertical	Specifies the vertical positioning data for objects in WordprocessingML documents. Default is absolute. Allowed values are: • absolute • top • center • bottom • inside • outside
mso-position-vertical-relative	Specifies relative vertical position data for objects in WordprocessingML documents. This modifies the mso-position-vertical property. Default is text. Allowed values are: • margin • page • text • line
mso-wrap-distance-bottom	Specifies the distance from the bottom of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
mso-wrap-distance-left	Specifies the distance from the left side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
mso-wrap-	Specifies the distance from the right side of the shape to the text

Attributes	Description
	`distance-right` — that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	`mso-wrap-distance-top` — Specifies the distance from the top of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	`mso-wrap-edited` — Specifies whether the wrap coordinates were customized by the user. If the wrap coordinates are generated by an editor, this property is `true`; otherwise they were customized by a user. Default is `false`.
	`mso-wrap-style` — Specifies the wrapping mode for text in shapes in WordprocessingML documents. Default is `square`. Allowed values are: • `square` - Wraps text inside the shape in a square. • `none` - Text does not wrap.
	`position` — Specifies the kind of positioning used to place an element. Default is `static`. When the element is contained inside a group, this property shall be `absolute`. Allowed values are: • `static` - The element is positioned according to the normal flow of the page. The `top` and `left` properties are ignored. If the object is anchored inline, this value is used. • `absolute` - The element is positioned relative to the parent, using the `top` and `left` properties. • `relative` - The element is positioned according to the normal flow of the page, but the `top` and `left` properties are used. The overlap of overlapping elements is governed by the `z-index` property.
	`rotation` — Specifies the angle that a shape is rotated, in degrees. Default is 0. Positive angles are clockwise.
	`top` — Specifies the position of the top of the containing block of the shape relative to the element above it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page.

Attributes	Description
	• <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
`visibility`	Specifies whether a shape is displayed. Only `inherit` and `hidden` are used; any other values are mapped to `inherit`. Default is `inherit`. Allowed values are: • `hidden` - The shape is not visible, but is still part of the flow of the objects in the browser. Mouse events are not processed. • `inherit` - The visibility state is inherited from the parent of the shape.
`width`	Specifies the width of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
`z-index`	Specifies the display order of overlapping shapes. Default is 0. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Uses the order that the shapes appear in the page, bottom to top. • <order>- A number that represents the stacking precedence. Shapes with higher numbers are placed on top of those with lower numbers. Negative numbers are allowed.

The following properties are only used by the textbox element (§19.1.2.22):

Property	Description
`direction`	Specifies the direction of the text in the textbox. Default is `ltr`. This property is superceded by the `mso-direction-alt` property if that is specified. Allowed values are:

Attributes	Description
	• `ltr` - Text is displayed left-to-right. • `rtl` - Text is displayed right-to-left.
`layout-flow`	Determines the flow of the text layout in a textbox. Default is `horizontal`. Allowed values are: • `horizontal` - Text is displayed horizontally. • `vertical` - Text is displayed vertically. • `vertical-ideographic` - Ideographic text is displayed vertically. • `horizontal-ideographic` - Ideographic text is displayed horizontally.
`mso-direction-alt`	Specifies an alternate direction for text in textboxes. Overrides the `direction` property. The only allowed value is `context`.
`mso-fit-shape-to-text`	Specifies whether the shape stretches to fit the text in the textbox. Default is `false`.
`mso-fit-text-to-shape`	Specifies whether the text stretches to fit the textbox. Default is `false`.
`mso-layout-flow-alt`	Specifies the alternate layout flow for text in textboxes. This property is used instead of `layout-flow` when the layout flow is from bottom to top for non-ideographic languages. Its only value is `bottom-to-top`.
`mso-next-textbox`	Specifies the ID of the next textbox in a series. Used to keep track of a set of linked textboxes. Default is no value.
`mso-rotate`	Specifies a specific rotation value for text in a textbox. Default is 0. Allowed values are: • 0 • 90 • 180 • -90
`mso-text-scale`	Specifies the scaling factor for fitting text to shapes. Default is 0. This property is only used if `mso-fit-text-to-shape` is `true`.
`v-text-anchor`	Specifies the vertical anchoring of text in a textbox. Default is `top`. The alignment of a text anchor only becomes evident if `mso-fit-text-to-shape` is `false`. This property is different from the `vertical-align` CSS property, which is used for ideographic languages. Allowed values are:

Attributes	Description

- `top`
- `middle`
- `bottom`
- `top-center`
- `middle-center`
- `bottom-center`
- `top-baseline`
- `bottom-baseline`
- `top-center-baseline`
- `bottom-center-baseline`

The following properties are only used by the textpath element (§19.1.2.23):

Property	Description
`font`	Specifies a compound value of font settings. Default is no value. The values are the same as those of the CSS font property. The order of definitions in the string is: font-style, font-variant, font-weight, font-size, line-height, font-family.
`font-family`	Specifies the family of the font. Default is no value. The values are the same as those of the CSS font-family property.
`font-size`	Specifies the size of the font. Default is no value. The font size is defined in points. The values are the same as those of the CSS font-size property.
`font-style`	Specifies the amount of slant for a font. Default is `normal`. The values are the same as those of the CSS font-style property. Allowed values are: • `normal` • `italic` • `oblique` - Treated the same as italic.
`font-variant`	Specifies the variant style of a font. Default is `normal`. The values are the same as those of the CSS font-variant property. Allowed values are: • `normal` • `small-caps`
`font-weight`	Specifies the thickness of the letters of the font. Default is `normal`. The values are the same as those of the CSS font-weight property. Allowed values are:

Value	Description

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>
<table>
<tr><td>normal
lighter
100
200
300
400</td><td>Treated as non-bold.</td></tr>
<tr><td>bold
bolder
500
600
700
800
900</td><td>Treated as bold.</td></tr>
</table>
</td></tr>
<tr><td>mso-text-shadow</td><td>Specifies whether a shadow is applied to the text on a text path. Default is false.</td></tr>
<tr><td>text-decoration</td><td>Specifies the style of text decoration. Default is none. The values are the same as those of the CSS text-decoration property. Allowed values are:
• none
• underline
• overline
• line-through
• blink</td></tr>
<tr><td>v-rotate-letters</td><td>Specifies whether the letters of the text are rotated counterclockwise by 90 degrees. Default is false.</td></tr>
<tr><td>v-same-letter-heights</td><td>Specifies whether all letters are the same height regardless of initial case. If true, the lowercase letters are stretched to the height of the uppercase letters. Default is false.</td></tr>
<tr><td>v-text-align</td><td>Specifies the alignment of text. Default is left. Allowed values are:
• left</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>

- right
- center
- justify
- letter-justify - Distributes the extra space between the letters.
- stretch-justify - Stretches the letters to fill in the space.

</td></tr>
<tr><td>v-text-kern</td><td>Specifies whether kerning is turned on. Default is false.</td></tr>
<tr><td>v-text-reverse</td><td>Specifies whether the layout order of rows is reversed. Default is false. This is used for vertical text layout.</td></tr>
<tr><td>v-text-spacing-mode</td><td>Specifies the mode for letter spacing. Default is tightening. This property determines whether space is removed between each letter (tightening) or added between each letter (tracking). The amount of letter spacing change is defined by the v-text-spacing property. Allowed values are:

• tightening
• tracking</td></tr>
<tr><td>v-text-spacing</td><td>Specifies the amount of spacing for text in 100ths of single line spacing. Default is 100.</td></tr>
</table>

The line (§19.1.2.12), polyline (§19.1.2.15) and curve (§19.1.2.3) elements ignore the following properties:

- top
- left
- width
- height

The following properties are not inherited by an element that references a shapetype element (§19.1.2.20) via the id attribute:

- flip
- height
- left
- margin-left
- margin-top
- position
- rotation
- top
- visibility
- width
</td></tr>
</table>

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>

- `z-index`

The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>target (Hyperlink Display Target)</td><td>

Specifies a frame or window that a URL is displayed in. Default is no value. Allowed values are:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><targetname></td><td>String containing the name of the frame or window in which to load the document.</td></tr>
<tr><td>`_blank`</td><td>Specifies that the linked document is loaded into a new blank window. This window is not named.</td></tr>
<tr><td>`_media`</td><td>Specifies that the linked document is loaded into the browser’s multimedia pane.</td></tr>
<tr><td>`_parent`</td><td>Specifies that the linked document is loaded into the immediate parent of the document containing the link.</td></tr>
<tr><td>`_search`</td><td>Specifies that the linked document is loaded into the browser's search pane.</td></tr>
<tr><td>`_self`</td><td>Specifies that the linked document is loaded into the window in which the link was clicked (the active window).</td></tr>
<tr><td>`_top`</td><td>Specifies that the linked document is loaded into the topmost window.</td></tr>
</table>

[*Example*:

```
<v:shape …
  href="http://www.openxmlformats.org"
  target="_self" … >
</v:shape>
```

end example]

The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>title (Shape Title)</td><td>

Specifies the text displayed when the mouse pointer moves over the shape. Default is no value.

[*Example*:

```
<v:shape … title="tooltip" … >
</v:shape>
```

end example]
</td></tr>
</table>

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

type (Shape Type Reference)	Specifies a reference to a shapetype ID that describes the standard path, fill and stroke properties of a shape. Properties specified in the shape override the shapetype properties. Default is no value.
	
	[*Example*: The following example defines a shapetype that is a simple rectangle and an actual shape instance that uses it and overrides the fill color.
	
	`<v:shapetype id="mytype"`
	`fillcolor="red" strokecolor="blue"`
	`coordorigin="0 0" coordsize="200 200"`
	`path="m 0,0 l 0,200, 200,200, 200,0 x e"/>`
	`</v:shapetype>`
	
	`<v:shape id="shape02" type="#mytype"`
	`fillcolor="green"`
	`style="position:relative;top:1;left:1;width:20;height:20">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

userdrawn (Exists In Master Slide)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the user has added the shape to a master slide. Default is `false`. Used by PresentationML.
	
	[*Example*:
	
	`<v:shape … o:userdrawn="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

userhidden (Hide Script Anchors)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a script anchor is hidden. Default is `false`. If `true`, script anchors stay hidden even if the shape is otherwise visible. A script anchor is the visual representation of a script that when displayed in an application.
	
	[*Example*:
	
	`<v:shape … o:userhidden="true" … >`
	`</v:shape>`

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

wrapcoords (Shape Bounding Polygon)	Specifies the bounding polygon that surrounds a shape. This is specified using a comma-delimited list of x and y coordinates: "x1,y1,x2,y2,x3,y3,..." This is used when text is tightly wrapped around a shape. Default is no value until the mso-wrap-mode style attribute is set to `tight` or `through`.
	
	[*Example*:
	
	`<v:shape …`
	`  wrapcoords="0,0 0,200, 200,200, 200,0" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Shape) is located in §A.6.1. *end note*]

19.1.2.20 shapetype (Shape Template)

This element defines a shape template that can be used to create other shapes. Shapetype is identical to the shape element (§19.1.2.19) except it cannot reference another shapetype element. The type attribute shall not be used with shapetype. Attributes defined in the shape override any that appear in the shapetype. CSS positioning attributes (such as `top`, `width`, `z-index`, `rotation`, `flip`) are not passed to a shape from a shapetype. To use this element, create a shapetype with a specific id attribute. Then create a shape and reference the shapetype's id using the type attribute.

[*Example*:

```
<v:shapetype id="mytype" fillcolor="silver" strokecolor="blue">
  <v:path v="m 0,0 l 0,1000, 1000,1000, 1000,0 x e"/>
  <v:fill on="true" type="gradient" color2="navy" angle="-45"/>
</v:shapetype>
<v:shape type="#mytype"
  style="position:absolute;top:10;left:10;width:50;height:50"/>
<v:shape type="#mytype" fillcolor="teal"
  style="position:absolute;top:10;left:75;width:75;height:50"/>
<v:shape type="#mytype"
  style="position:absolute;top:10;left:165;width:50;height:50">
```

```
    <v:fill type="solid"/>
  </v:shape>
  <v:shape type="#mytype" path="m 500,0 l 1000,1000 0,1000 x e"
  style="position:absolute;top:10;left:230;width:50;height:50"/>
```

end example]

Attributes	Description

adj (Adjustment Parameters)	Specifies a comma-delimited list of parameters, or adjustment values, used to define values for a parameterized formula. Values can be omitted. There can be up to 8 adjust values. Each value is referenced using # followed by a number corresponding to the zero-based index for that value in the list of adjustment values. [*Example*: For example, #2 references the second value in the adj list. *end example*]
	
	[*Example*: The following shape uses formulas to define a simple rectangle. The adj values are referenced by the eqn attribute of the f element (§19.1.2.4) and in turn referenced by the path element (§19.1.2.14).
	
	`<v:shape`
	`  coordorigin="0 0" coordsize="200 200"`
	`  style="position:relative;top:30;left:30;width:20;height:20"`
	`  adj="1, 1, 1, 200, 200, 200, 200, 1">`
	`  <v:path v="m @0,@1 l @2,@3, @4,@5, @6,@7 x e"/>`
	`  <v:formulas>`
	`  <v:f eqn="val #0"/>`
	`  <v:f eqn="val #1"/>`
	`  <v:f eqn="val #2"/>`
	`  <v:f eqn="val #3"/>`
	`  <v:f eqn="val #4"/>`
	`  <v:f eqn="val #5"/>`
	`  <v:f eqn="val #6"/>`
	`  <v:f eqn="val #7"/>`
	`  </v:formulas>`
	`</v:shape>`
	
	This is the equivalent of:
	
	`<v:shape`
	`  coordorigin="0 0" coordsize="200 200"`
	`  style="position:relative;top:30;left:30;width:20;height:20"`
	`  path="m 1,1 l 1,200, 200,200, 200,1 x e">`
	`</v:shape>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

allowincell (Allow in Table Cell)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can be placed in a table. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:allowincell="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

allowoverlap (Allow Shape Overlap)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape can overlap another shape. Default is `true`. If `false`, the shape shifts left or right so as not to overlap another shape, similar to the behavior of the HTML float attribute.
	
	[*Example*:
	
	`<v:shape … o:allowoverlap="false" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

alt (Alternate Text)	Specifies alternative text describing the graphical object. This text should provide a brief description of the shape for use by accessibility tools. Default is no value.
	
	[*Example*: The alt text describes the basic shape:
	
	`<v:shape … fillcolor="red"`
	`  alt="Red rectangle">`
	`</v:shape>`
	
	The alt text describes the contents of a shape displaying an image:
	
	`<v:shape … alt="Picture of a sunset">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

Attributes	Description

borderbottomcolor (Bottom Border Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the bottom border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderbottomcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderleftcolor (Border Left Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the left border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderleftcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

borderrightcolor (Border Right Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the right border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:borderrightcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

bordertopcolor (Border Top Color)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the top border color of an inline shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … o:bordertopcolor="red" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

bullet (Graphical	Specifies whether the shape is a graphical bullet. Default is `false`.

Attributes	Description

Bullet)
 	
Namespace: urn:schemas-microsoft-com:office:office
	[*Example*:
	
	`<v:shape … o:bullet="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

button (Button Behavior Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape exhibits button press behavior on click. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:button="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

bwmode (Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies how a shape renders for black-and-white output devices. When a shape is printed on a black-and-white printer or displayed in a black-and-white view in an application, several options are possible. Default is `auto`, which uses o:bwnormal for normal black-and-white rendering and o:bwpure for pure black-and-white rendering.
	
	bwnormal and bwpure are subordinate to bwmode. If bwmode is "auto" then the value for bwnormal or bwpure is used depending on what the output format is. An application can define for itself what, if any, difference there is between normal B&W and pure B&W. [*Example*: Normal B&W might allow greyscale and pure B&W might not. *end example*]
	[*Example*: This shape renders in grayscale in a black-and-white environment:
	
	`<v:shape … o:bwmode="grayscale" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

bwnormal (Normal Black-and-White Mode) Namespace:	Specifies the black-and-white mode for normal black-and-white output devices. Default is `auto`. [*Example*: This shape renders in a pale grayscale in a normal black-and-white environment:

Attributes	Description

urn:schemas-microsoft-com:office:office	`<v:shape … o:bwmode="auto" o:bwnormal="lightgrayscale" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

bwpure (Pure Black-and-White Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the black-and-white mode for pure black-and-white output devices. Default is `auto`.
	
	[*Example*: This shape renders in high contrast when in a pure black-and-white environment:
	
	`<v:shape … o:bwmode="auto" o:bwpure="highcontrast" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_BWMode simple type (§19.2.3.3).

chromakey (Image Transparency Color)	Specifies a color value that is transparent and show anything behind the shape. Default is no value.
	
	[*Example*:
	
	`<v:image … chromakey="white" …>`
	`</v:image>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

class (CSS Reference)	Specifies a reference to the definition of a CSS style. Default is no value.
	
	[*Example*: The snippets below are equivalent:
	
	`…`
	`.narrowstyle {width:50;height:100}`
	`…`
	`<v:shape … class="narrowstyle"`
	`  style="top:1;left:1">`
	`</v:shape>`

Attributes	Description

	`<v:shape … style="top:1;left:1;`
	`  width:50;height:100">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

clip (Clipping Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the clipping region is active. This is used in conjunction with the clippath (§19.2.2.3) element to create a clipping region.
	
	[*Example*:
	
	`<v:shape … o:clip="true">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

cliptowrap (Clip to Wrapping Polygon)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the clipping region for the shape aligns with the wrapping polygon that tightly surrounds the entire shape (essentially, that the shape shall not be drawn beyond its wrapping polygon's extents – if it does, the shape shall be clipped). Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:cliptowrap="true">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

connectortype (Shape Connector Type)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the kind of connector used for joining shapes. Default is `straight`.
	
	[*Example*:
	
	`<v:shape … o:connectortype="elbow" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ConnectorType simple type (§19.2.3.7).

coordorigin	Specifies the coordinate of the top left corner of the shape's coordinate space. This

Attributes	Description

(Coordinate Space Origin)	determines the position of the (0,0) coordinate space origin within the shape's bounding box. Default is "0,0", which places the (0,0) origin at the top left corner of the bounding box.
	
	This affects shape properties that specify coordinate positions, such as the path attribute. Thus a path can be defined against a generic (0,0) origin and the coordorigin value translates the entire path within the shape's bounding space.
	
	[*Example*: The horizontal and vertical coordinate space ranges from -100 to +100 because the coordinate space (coordsize) is 200 by 200 and the top left coordinate is (-100,-100). The (0,0) origin lies at the center of the shape's bounding box, as evidenced by the position of the shape's path within the coordinate space:
	
	`<v:shape … coordsize="200,200"`
	`  coordorigin="-100,-100"`
	`  path="m 0,0 l 0,50, 50,50, 50,0 x e">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

coordsize (Coordinate Space Size)	Specifies the size of the shape's coordinate space in coordinate units. Default is "1000,1000".
	
	The physical size of a coordinate unit length is determined by both the size of the coordinate space (coordsize) and the size of the shape (style width and height). The coordsize attribute defines the number of horizontal and vertical subdivisions into which the shape's bounding box is divided. The combination of coordsize and style width/height effective scales the shape anisotropically.
	
	[*Example*: The path is 50 units wide and tall, which is 25% of the size of the coordinate space:
	
	`<v:shape … coordsize="200,200"`
	`  coordorigin="-100,-100"`
	`  path="m 0,0 l 0,50, 50,50, 50,0 x e">`
	`</v:shape>`

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema string datatype.

dgmlayout
(Diagram Node Layout Identifier)
 	
Namespace:
urn:schemas-microsoft-com:office:office
	Specifies the kind of automatic layout to apply to the diagram node. This is only meaningful if the shape is a node in an organization chart, which is denoted by the `orgchart` value of the editas attribute of the group element.
	[*Example*:
	
	`<v:shape … dgmlayout="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).

dgmlayoutmru
(Diagram Node Recent Layout Identifier)
 	
Namespace:
urn:schemas-microsoft-com:office:office
	Specifies the kind of automatic layout most recently used on the child elements of the diagram node. This is only meaningful if the shape is a node in anorganization chart, which is denoted by the `orgchart` value of the editas attribute of the group element.
	
	[*Example*:
	
	`<v:shape … dgmlayout="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_DiagramLayout simple type (§19.2.3.10).

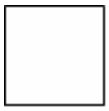
dgmnodekind
(Diagram Node Identifier)
 	
Namespace:
urn:schemas-microsoft-com:office:office
	Specifies an optional, application-defined parameter that is intended to be used by the application to tag different types of nodes in a diagram.
	
	[*Example*:
	
	`<v:shape … dgmnodekind="1">`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

doubleclicknotify (Double-click	Specifies that an event message is sent when a shape is double-clicked. Default is `false`.

Attributes	Description

Notification Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	[*Example*:
	
	`<v:shape … o:doubleclicknotify="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

fillcolor (Fill Color)	Specifies the color to use for the fill. Default is `white`. If the fill element (§19.1.2.5) is present, its color attribute takes precedence. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*: This shape is red if its fill is visible:
	
	`<v:shape … fillcolor="red" … >`
	`</v:shape>`
	
	This is equivalent to:
	
	`<v:shape … fillcolor="#ff0000" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

filled (Shape Fill Toggle)	Specifies whether the closed path is filled. Default is `true`. This attribute is overridden by the fill on attribute.
	
	[*Example*:
	
	`<v:shape … filled="f"`
	`  fillcolor="red" …>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

Attributes	Description

forcedash (Force Dashed Outline)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a dashed outline is used to draw a shape when a shape has no line or fill. Default is `false`.
	
	Used by PresentationML placeholders to draw a dashed outline when there is no line and no fill for a shape.
	
	[*Example*:
	
	`<v:shape … o:forcedash="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hr (Horizontal Rule Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that a shape is a horizontal rule. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hr="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hralign (Horizontal Rule Alignment)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the alignment of a horizontal rule. Default is `left`.
	
	[*Example*:
	
	`<v:shape … o:hralign="center" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_HrAlign simple type (§19.2.3.16).

href (Hyperlink Target)	Specifies a hyperlink URL target for the shape. Default is no value.
	
	[*Example*:
	
	`<v:shape … href="http://www.openxmlformats.org" … >`
	`</v:shape>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

hrnoshade (Horizontal Rule 3D Shading Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies that the horizontal rule does not have 3-D shading. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hrnoshade="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

hrpct (Horizontal Rule Length Percentage)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the length of a horizontal rule as a percentage of page width. Default is 0.
	
	[*Example*:
	
	`<v:shape … o:hrpct="85" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema float datatype.

hrstd (Horizontal Rule Standard Display Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a shape is displayed as a standard horizontal rule. Only applies if hr is `true`. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:hrstd="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

id (Unique Identifier)	Specifies a unique identifier that can be used to reference a VML object.
	
	Default is no value.
	
	[*Example*:
	
	`<v:shape … id="myShape" … >`
	`</v:shape>`

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema string datatype.

insetmode (Text Inset Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the application calculates the internal text margin instead of using the inset attribute. Default is `custom`. This attribute is only meaningful for text boxes.
	
	[*Example*:
	
	`<v:shape … o:insetmode="auto" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_InsetMode simple type (§19.2.3.17).

insetpen (Inset Border From Path)	Specifies that the border shall be displayed inside of the path defining the shape, rather than along the path (the default border placement), or outside of the path as might be done with an image.
	
	[*Example*:
	
	`<v:shape … insetpen="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

master (Master Element Toggle) Namespace: urn:schemas-microsoft-com:office:office	Specifies whether the shapetype is a master element. If `true`, it is rendered by the rendering engine. Default is `false`. The possible values for this attribute are defined by the W3C XML Schema string datatype.

ole (Embedded Object Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the shape is an embedded object. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:ole="true" … >`
	`</v:shape>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

oleicon (Embedded Object Icon Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether an embedded object is displayed as an icon. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oleicon="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

oned (Shape Handle Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the extra handles of a shape are hidden. If `true`, hides all shape handles except the top left and bottom right; that is, the same handles that are used for a straight line segment. Default is `false`.
	
	[*Example*:
	
	`<v:shape … o:oned="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

opacity (Fill Color Opacity)	Specifies the opacity of the primary fill color. Default is 1.0. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied. [*Example*: A value of "52429f" represents 52429/65536 or 0.8. *end example*]
	
	[*Example*: The red color is 25% opaque:
	
	`<v:fill type="gradient" color="red"`
	`color2="blue" opacity=".25">`
	`</v:fill>`
	
	  `opacity="1"`
	  `opacity=".25"`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.
path (Edge Path)	Specifies the line that makes up the edges of a shape. See the v attribute of the path element (§19.1.2.14) for a full description. The possible values for this attribute are defined by the W3C XML Schema string datatype.

preferrelative (Relative Resize Toggle)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the original size of an object is saved after reformatting. Default is `false`. If `true`, the original size of the object is stored and all resizing is based on a percentage of that original size. Otherwise, each resizing resets the scale to 100%.
	
	[*Example*:
	
	`<v:shape … o:preferrelative="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

print (Print Toggle)	Specifies whether the shape is printed. Default is `true`.
	
	[*Example*:
	
	`<v:shape … print="false" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

regroupid (Regroup ID)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies a previous group for a shape. An ID number is used to identify groups of shapes that are no longer grouped. This allows shapes to be regrouped programmatically.
	
	[*Example*: The shape was part of a group identified by the ID 040754:
	
	`<v:shape … o:regroupid="040754" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

spid (Optional	Specifies an optional string that an application can use to Identify the particular shape.

Attributes	Description
String) Namespace: urn:schemas-microsoft-com:office:office	Default is no value. The possible values for this attribute are defined by the W3C XML Schema string datatype.
spt (Optional Number) Namespace: urn:schemas-microsoft-com:office:office	Specifies an optional number that an application can use to associate the particular shape with a defined shape type. Default is 0. The possible values for this attribute are defined by the W3C XML Schema float datatype.

strokecolor (Shape Stroke Color)	Specifies the primary color of the brush to use to stroke the path of the shape. Default is `black`. The color attribute of the stroke element (§19.1.2.21) overrides this. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*:
	
	`<v:shape … strokecolor="red" …>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

stroked (Shape Stroke Toggle)	Specifies whether the path defining the shape is stroked with a solid line. The stroke element (§19.1.2.21) defines other strokes. The on attribute of the stroke element overrides this attribute. Default is `true`.
	
	[*Example*:
	
	`<v:shape … fillcolor="red"`
	`  stroked="false" strokecolor="blue"…>`
	`</v:shape>`
	
	 

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

strokeweight (Shape Stroke Weight)	Specifies the width of the brush to use to stroke the path. Default is 1 point. If a number is given without units, the emu is used. The weight attribute of the stroke element (§19.1.2.21) overrides this attribute.
	
	[*Example*:
	
	`<v:shape … strokeweight="3pt" … >`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

style (Shape Styling Properties)	Specifies the CSS2 styling properties of the shape. The CSS2 (Cascading Style Sheets, Level 2) specification, a Recommendation of the World Wide Web Consortium, is available here: http://www.w3.org/TR/REC-CSS2.
	
	This attribute uses a semi-colon delimited list of "name:value" pairs, the syntax defined by section 4.1.8 of the CSS2 specification for grouped declarations (without the surrounding braces). The following tables define the allowed properties and the VML treatment of each.
	
	[*Example*:
	`<v:shape … style='position:absolute;width:100pt;height:50pt' … >`
	`</v:shape>`
	end example]
	
	Property / **Description**
	`flip` — Specifies that the orientation of a shape is flipped. Default is no value. Allowed values are:
	• x - Flip along the y-axis, reversing the x-coordinates.
	• y - Flip along the x-axis, reversing the y-coordinates.
	• xy - Flip along both the y- and x-axis.
	• yx - Flip along both the x- and y-axis.

Attributes	Description	
	`height`	Specifies the height of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`left`	Specifies the position of the left of the containing block of the shape relative to the element left of it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-bottom`	Specifies the position of the bottom of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`margin-left`	Specifies the position of the left of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm,

Attributes	Description	
		mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-right`	Specifies the position of the right of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-top`	Specifies the position of the top of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`mso-position-horizontal`	Specifies the horizontal positioning data for objects in WordprocessingML documents. Default is `absolute`. Allowed values are: • `absolute` • `left` • `center` • `right` • `inside` • `outside`
	`mso-position-horizontal-relative`	Specifies relative horizontal position data for objects in WordprocessingML documents. This modifies the `mso-position-horizontal` property. Default is `text`. Allowed values are:

Attributes	Description
	• margin • page • text • char
mso-position-vertical	Specifies the vertical positioning data for objects in WordprocessingML documents. Default is absolute. Allowed values are: • absolute • top • center • bottom • inside • outside
mso-position-vertical-relative	Specifies relative vertical position data for objects in WordprocessingML documents. This modifies the mso-position-vertical property. Default is text. Allowed values are: • margin • page • text • line
mso-wrap-distance-bottom	Specifies the distance from the bottom of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
mso-wrap-distance-left	Specifies the distance from the left side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
mso-wrap-distance-right	Specifies the distance from the right side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
mso-wrap-distance-top	Specifies the distance from the top of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the

Attributes	Description
	origin.
`mso-wrap-edited`	Specifies whether the wrap coordinates were customized by the user. If the wrap coordinates are generated by an editor, this property is `true`; otherwise they were customized by a user. Default is `false`.
`mso-wrap-style`	Specifies the wrapping mode for text in shapes in WordprocessingML documents. Default is `square`. Allowed values are: • `square` - Wraps text inside the shape in a square. • `none` - Text does not wrap.
`position`	Specifies the kind of positioning used to place an element. Default is `static`. When the element is contained inside a group, this property shall be `absolute`. Allowed values are: • `static` - The element is positioned according to the normal flow of the page. The `top` and `left` properties are ignored. If the object is anchored inline, this value is used. • `absolute` - The element is positioned relative to the parent, using the `top` and `left` properties. • `relative` - The element is positioned according to the normal flow of the page, but the `top` and `left` properties are used. The overlap of overlapping elements is governed by the `z-index` property.
`rotation`	Specifies the angle that a shape is rotated, in degrees. Default is 0. Positive angles are clockwise.
`top`	Specifies the position of the top of the containing block of the shape relative to the element above it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
`visibility`	Specifies whether a shape is displayed. Only `inherit` and `hidden` are used; any other values are mapped to `inherit`. Default is `inherit`. Allowed values are:

Attributes	Description

	• `hidden` - The shape is not visible, but is still part of the flow of the objects in the browser. Mouse events are not processed. • `inherit` - The visibility state is inherited from the parent of the shape.
`width`	Specifies the width of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
`z-index`	Specifies the display order of overlapping shapes. Default is 0. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Uses the order that the shapes appear in the page, bottom to top. • <order>- A number that represents the stacking precedence. Shapes with higher numbers are placed on top of those with lower numbers. Negative numbers are allowed.

The following properties are only used by the textbox element (§19.1.2.22):

Property	Description
`direction`	Specifies the direction of the text in the textbox. Default is `ltr`. This property is superceded by the `mso-direction-alt` property if that is specified. Allowed values are: • `ltr` - Text is displayed left-to-right. • `rtl` - Text is displayed right-to-left.
`layout-flow`	Determines the flow of the text layout in a textbox. Default is `horizontal`. Allowed values are: • `horizontal` - Text is displayed horizontally. • `vertical` - Text is displayed vertically.

Attributes	Description
	• `vertical-ideographic` - Ideographic text is displayed vertically. • `horizontal-ideographic` - Ideographic text is displayed horizontally.
`mso-direction-alt`	Specifies an alternate direction for text in textboxes. Overrides the `direction` property. The only allowed value is `context`.
`mso-fit-shape-to-text`	Specifies whether the shape stretches to fit the text in the textbox. Default is `false`.
`mso-fit-text-to-shape`	Specifies whether the text stretches to fit the textbox. Default is `false`.
`mso-layout-flow-alt`	Specifies the alternate layout flow for text in textboxes. This property is used instead of `layout-flow` when the layout flow is from bottom to top for non-ideographic languages. Its only value is `bottom-to-top`.
`mso-next-textbox`	Specifies the ID of the next textbox in a series. Used to keep track of a set of linked textboxes. Default is no value.
`mso-rotate`	Specifies a specific rotation value for text in a textbox. Default is 0. Allowed values are: • 0 • 90 • 180 • -90
`mso-text-scale`	Specifies the scaling factor for fitting text to shapes. Default is 0. This property is only used if `mso-fit-text-to-shape` is `true`.
`v-text-anchor`	Specifies the vertical anchoring of text in a textbox. Default is `top`. The alignment of a text anchor only becomes evident if `mso-fit-text-to-shape` is `false`. This property is different from the `vertical-align` CSS property, which is used for ideographic languages. Allowed values are: • `top` • `middle` • `bottom` • `top-center` • `middle-center` • `bottom-center` • `top-baseline`

Attributes	Description
	• bottom-baseline • top-center-baseline • bottom-center-baseline

The following properties are only used by the textpath element (§19.1.2.23):

Property	Description
font	Specifies a compound value of font settings. Default is no value. The values are the same as those of the CSS font property. The order of definitions in the string is: font-style, font-variant, font-weight, font-size, line-height, font-family.
font-family	Specifies the family of the font. Default is no value. The values are the same as those of the CSS font-family property.
font-size	Specifies the size of the font. Default is no value. The font size is defined in points. The values are the same as those of the CSS font-size property.
font-style	Specifies the amount of slant for a font. Default is normal. The values are the same as those of the CSS font-style property. Allowed values are: • normal • italic • oblique - Treated the same as italic.
font-variant	Specifies the variant style of a font. Default is normal. The values are the same as those of the CSS font-variant property. Allowed values are: • normal • small-caps
font-weight	Specifies the thickness of the letters of the font. Default is normal. The values are the same as those of the CSS font-weight property. Allowed values are:

Value	Description
normal lighter 100 200	Treated as non-bold.

Attributes	Description
	<table><tr><td></td><td><table><tr><td>300 400</td><td></td></tr><tr><td>bold bolder 500 600 700 800 900</td><td>Treated as bold.</td></tr></table></td></tr><tr><td>mso-text-shadow</td><td>Specifies whether a shadow is applied to the text on a text path. Default is false.</td></tr><tr><td>text-decoration</td><td>Specifies the style of text decoration. Default is none. The values are the same as those of the CSS text-decoration property. Allowed values are: • none • underline • overline • line-through • blink</td></tr><tr><td>v-rotate-letters</td><td>Specifies whether the letters of the text are rotated counterclockwise by 90 degrees. Default is false.</td></tr><tr><td>v-same-letter-heights</td><td>Specifies whether all letters are the same height regardless of initial case. If true, the lowercase letters are stretched to the height of the uppercase letters. Default is false.</td></tr><tr><td>v-text-align</td><td>Specifies the alignment of text. Default is left. Allowed values are: • left • right • center • justify • letter-justify - Distributes the extra space between the letters. • stretch-justify - Stretches the letters to fill in the space.</td></tr></table>

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>v-text-kern</td><td>Specifies whether kerning is turned on. Default is false.</td></tr>
<tr><td>v-text-reverse</td><td>Specifies whether the layout order of rows is reversed. Default is false. This is used for vertical text layout.</td></tr>
<tr><td>v-text-spacing-mode</td><td>Specifies the mode for letter spacing. Default is tightening. This property determines whether space is removed between each letter (tightening) or added between each letter (tracking). The amount of letter spacing change is defined by the v-text-spacing property. Allowed values are:

• tightening
• tracking</td></tr>
<tr><td>v-text-spacing</td><td>Specifies the amount of spacing for text in 100ths of single line spacing. Default is 100.</td></tr>
</table>

The line (§19.1.2.12), polyline (§19.1.2.15) and curve (§19.1.2.3) elements ignore the following properties:

• top
• left
• width
• height

The following properties are not inherited by an element that references a shapetype element (§19.1.2.20) via the id attribute:

• flip
• height
• left
• margin-left
• margin-top
• position
• rotation
• top
• visibility
• width
• z-index

The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>target (Hyperlink Display Target)</td><td>Specifies a frame or window that a URL is displayed in. Default is no value. Allowed values are:</td></tr>
</table>

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><targetname></td><td>String containing the name of the frame or window in which to load the document.</td></tr>
<tr><td>_blank</td><td>Specifies that the linked document is loaded into a new blank window. This window is not named.</td></tr>
<tr><td>_media</td><td>Specifies that the linked document is loaded into the browser’s multimedia pane.</td></tr>
<tr><td>_parent</td><td>Specifies that the linked document is loaded into the immediate parent of the document containing the link.</td></tr>
<tr><td>_search</td><td>Specifies that the linked document is loaded into the browser's search pane.</td></tr>
<tr><td>_self</td><td>Specifies that the linked document is loaded into the window in which the link was clicked (the active window).</td></tr>
<tr><td>_top</td><td>Specifies that the linked document is loaded into the topmost window.</td></tr>
</table>

[Example:

<pre>
<v:shape …
  href="http://www.openxmlformats.org"
  target="_self" … >
</v:shape>
</pre>

end example]

The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>title (Shape Title)</td><td>
Specifies the text displayed when the mouse pointer moves over the shape. Default is no value.

[Example:

<pre>
<v:shape … title="tooltip" … >
</v:shape>
</pre>

end example]

The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>userdrawn (Exists In Master Slide)

Namespace:</td><td>
Specifies whether the user has added the shape to a master slide. Default is <code>false</code>. Used by PresentationML.

[Example:
</td></tr>
</table>

Attributes	Description
urn:schemas-microsoft-com:office:office	<v:shape … o:userdrawn="true" … > </v:shape> *end example*] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
userhidden (Hide Script Anchors) Namespace: urn:schemas-microsoft-com:office:office	Specifies whether a script anchor is hidden. Default is `false`. If `true`, script anchors stay hidden even if the shape is otherwise visible. A script anchor is the visual representation of a script that when displayed in an application. [*Example*: <v:shape … o:userhidden="true" … > </v:shape> *end example*] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
wrapcoords (Shape Bounding Polygon)	Specifies the bounding polygon that surrounds a shape. This is specified using a comma-delimited list of x and y coordinates: "x1,y1,x2,y2,x3,y3,…" This is used when text is tightly wrapped around a shape. Default is no value until the mso-wrap-mode style attribute is set to `tight` or `through`. [*Example*: <v:shape … wrapcoords="0,0 0,200, 200,200, 200,0" … > </v:shape> *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Shapetype) is located in §A.6.1. *end note*]

19.1.2.21 stroke (Line Stroke Settings)

This element describes how to draw the path if something beyond solid line with a solid color is desired. The attributes of the stroke element can be used to describe a powerful set of stroke properties. Extensions to the VML stroke definition are encoded as sub-elements of stroke.

[*Example*:

```
<v:polyline points="0pt,0pt,50pt,0pt,50pt,35pt,15pt,35pt,
  15pt,15pt,75pt,15pt">
  <v:stroke startarrow="classic" endarrow="classic"
    startarrowwidth="wide" endarrowwidth="wide" dashstyle="dashdot"
    weight="2pt" color="teal" linestyle="thinThin"/>
</v:polyline>
```

end example]

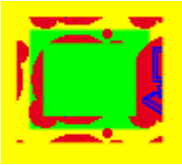
Attributes	Description

althref (Alternate Image Reference)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies an alternate reference for an image in Macintosh PICT format.
	
	[*Example*:
	
	`<v:stroke … althref="myimage.pcz" … >`
	`</v:stroke>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

color (Stroke Color)	Specifies the stroke color. Overrides the strokecolor attribute of a shape. Default is `black`. See the fillcolor attribute for a list of supported named colors.
	
	[*Example*: The shape stroke is blue:
	
	`<v:shape … strokecolor="red" … >`
	`  <v:stroke color="blue"/>`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

color2 (Stroke Alternate Pattern Color)	Specifies a second color for strokes, used when filltype is `pattern`. Default is no value. When a pattern fill is used for the stroke, the stroke color is used in colored parts of the source image. The color2 defines an alternate color to use in place of black in the source image.

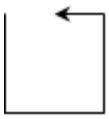
Attributes	Description

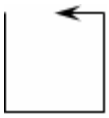
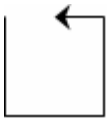
	[*Example*: This unusual example is intended to demonstrate how the image and colors interact to create a patterned stroke. The yellow background shows transparency. The non-square shape and square image create an effective offset. The heavy stroke weight shows more of the image. The green shape fill shows how the stroke is overlaid on the shape.
	
	`<v:background fillcolor="yellow"/>`
	`<v:shape style="width:60;height:50"`
	`  strokecolor="red" fillcolor="lime"`
	`  path="m 0,0 l 0,1000 1000,1000 1000,0 x e">`
	`  <v:stroke filltype="pattern" weight="10pt"`
	`  src="myimage.gif" color2="blue"/>`
	`</v:shape>`
	
	  , where myimage.gif is: 
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

dashstyle (Stroke Dash Pattern)	Specifies the dot and dash pattern for a stroke. Default is `solid`. Pre-defined values are: • `solid` • `shortdash` • `shortdot` • `shortdashdot` • `shortdashdotdot` • `dot` • `dash` • `longdash` • `dashdot` • `longdashdot` • `longdashdotdot` A custom-defined dash pattern can also be specified using a series of numbers. These define the length of the dash (the drawn part of the stroke) and the length of the space between the dashes. The lengths are relative to the line width: a length of 1 is equal to the line width. The endcap style is applied to each dash but the arrow style is not. The string defines the length of the dash then the length of the space. This can be repeated to form complex dash styles. The string should always contain a pair of numbers; if it contains an odd number of numbers the last is disregarded. 0 implies a dot that is

Attributes	Description

	fourfold symmetrical (with round end caps, this is a circle).
	
	[*Example*:
	
	`<v:stroke dashstyle="0 2" weight="3pt"`
	`  endcap="round">`
	`</v:stroke>`
	
	 
	
	`<v:stroke dashstyle="longdashdotdot"`
	`  weight="2pt">`
	`</v:stroke>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

endarrow (Line End Arrowhead)	Specifies an arrowhead for the end of a line. Default is none. The path shall not be closed with the x command for the arrowhead to show. Allowed values are: • `none` • `block` • `classic` • `diamond` • `oval` • `open` [*Example*: `<v:stroke endarrow="classic"/>`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowType simple

Attributes	Description
	type (§19.1.3.8).
endarrowlength (Line End Arrowhead Length)	Specifies the length of the arrowhead at the end of a line. Default is `medium`. Allowed values are: • `short` • `medium` • `long` [*Example*: `<v:stroke … endarrowlength="long" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowLength simple type (§19.1.3.7).
endarrowwidth (Line End Arrowhead Width)	Specifies the width of the arrowhead at the end of a line. Default is `medium`. Allowed values are: • `narrow` • `medium` • `wide` [*Example*: `<v:stroke … endarrowwidth="wide" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowWidth simple type (§19.1.3.9).
endcap (Line End Cap)	Specifies the cap style for the end of a stroke. Default is `flat`. Allowed values are: • `flat` • `square`

Attributes	Description
	• `round` [*Example*: `<v:stroke … endcap="round" weight="10pt" … />`  `endcap="flat"`  `endcap="square"`  `endcap="round"` *end example*] The possible values for this attribute are defined by the ST_StrokeEndCap simple type (§19.1.3.10).

filltype (Stroke Image Style)	Specifies the kind offill used for the background of a stroke. Default is `solid`. Allowed values are:
	
	• `solid` - The fill pattern is solid.
	• `tile` - The fill image is tiled.
	• `pattern` - The fill image is stretched to form a pattern.
	• `frame` - The fill image becomes a border for the shape.
	
	[*Example*:
	
	`<v:shape style="width:50;height:50"`
	`  strokecolor="red"`
	`  path="m 0,0 l 0,1000 1000,1000 1000,0 x e">`
	`  <v:stroke filltype="frame" weight="10pt"`
	`  src="border.gif"/>`
	`</v:shape>`
	
	  , where border.gif is: 
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_FillType simple type (§19.1.3.4).

forcedash (Force Dashed Outline)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether a dashed outline is used to draw a shape when a shape has no line or fill. Default is `false`.
	
	Used by PresentationML placeholders to draw a dashed outline when there is no line and no fill for a shape.
	
	[*Example*:
	
	`<v:shape … o:forcedash="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

href (Original Image Reference)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the URL to the original image file. Used only if the picture has been linked and embedded. Default is no value.
	
	[*Example*:
	
	`<v:fill … o:href="myimage.gif" … >`
	`</v:fill>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

id (Relationship) Namespace: …/officeDocument/2006/relationships	Specifies the relationship ID of the relationship to the image used for the stroke. The specified relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/image or the document shall be considered non-conformant. [*Example*: The markup specifies the associated relationship part with relationship ID `rId10` contains the corresponding relationship information: `< … r:id="rId10" />` *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).
id (Unique Identifier)	Specifies a unique identifier that can be used to reference a VML object.

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>Default is no value.

[Example:

<pre><v:shape … id="myShape" … >
</v:shape></pre>
end example]

The possible values for this attribute are defined by the W3C XML Schema string datatype.</td></tr>
<tr><td>imagealignshape (Stoke Image Alignment)</td><td>Specifies the alignment of the stroke image. If <code>true</code>, the image is aligned with the shape. Otherwise, it is aligned with the containing scope. Default is <code>true</code>.

[Example: The top position offset shifts the image alignment relative to the containing window:

<pre><v:shape fillcolor="silver"
  style="top:20;width:50;height:50"
  path="m 0,0 l 0,1000 1000,1000 1000,0 x e">
  <v:stroke imagealignshape="false" weight="20pt"
  filltype="tile" src="myimage.gif"/>
</v:shape></pre>
  <code>imagealignshape="false"</code>

  <code>imagealignshape="false"</code>

end example]

The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).</td></tr>
<tr><td>imageaspect (Stroke Image Aspect Ratio)</td><td>Specifies how the stroke image aspect ratio is preserved. Default is <code>ignore</code>. Allowed values are:

<table><tr><td>Value</td><td>Description</td></tr><tr><td><code>ignore</code></td><td>Ignore aspect issues.</td></tr></table></td></tr>
</table>

Attributes	Description

	<table><tr><td>atleast</td><td>Image is at least as big as imagesize.</td></tr><tr><td>atmost</td><td>Image is no bigger than imagesize.</td></tr></table>
	[*Example*:
	
	`<v:stroke filltype="frame" weight="10pt"`
	`src="border.gif" imagealignshape="true"`
	`imageaspect="atleast">`
	`</v:stroke>`
	
	 
	`imagealignshape="ignore"`
	
	`imagealignshape="atleast"`
	
	`imagealignshape="atmost"`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ImageAspect simple type (§19.1.3.5).

imagesize (Stroke Image Size)	Specifies the size of the image for the stroke. Default is the size of the image. [*Example*: `<v:stroke … imagesize="10pt,10pt" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

insetpen (Inset Border From Path)	Specifies that the border shall be displayed inside of the path defining the shape, rather than along the path (the default border placement), or outside of the path as might be done with an image.
	
	[*Example:*
	
	`<v:shape … insetpen="true" … >`
	`</v:shape>`

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

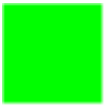
joinstyle (Line End Join Style)	Specifies the join style for line ends. Default is `round`.
	
	• `round`
	• `bevel`
	• `miter`
	
	[*Example*:
	
	`<v:polyline strokeweight="10pt" strokecolor="navy"`
	`  points="10pt,10pt,50pt,50pt,90pt,10pt">`
	`  <v:stroke joinstyle="bevel"/>`
	`</v:polyline>`
	
	 
	`joinstyle="round"`
	`joinstyle="bevel"`
	`joinstyle="miter"`
	
	end example]
	
	The possible values for this attribute are defined by the ST_StrokeJoinStyle simple type (§19.1.3.11).

linestyle (Stroke Line Style)	Specifies the line style of the stroke. Default is `single`. • `single` • `thinThin` • `thinThick` • `thickThin` • `thickBetweenThin` [*Example*:

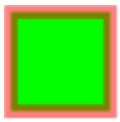
Attributes	Description

	`<v:stroke linestyle="thickThin" weight="5pt">`
	`</v:stroke>`
	 
	end example]
	
	The possible values for this attribute are defined by the ST_StrokeLineStyle simple type (§19.1.3.12).

miterlimit (Miter Joint Limit)	Specifies the smoothness of the miter joint, or the maximum distance between the inner point and outer point of a joint. This number is a multiple of the thickness of the line. Default is 8.
	
	[*Example*:
	
	`<v:stroke joinstyle="miter" weight="10pt"`
	`  miterlimit="2">`
	`</v:stroke>`
	 
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema decimal datatype.

on (Stroke Toggle)	Specifies whether the stroke is displayed. Default is `true`. This attribute overrides the shape's stroke attribute.
	
	[*Example:*
	
	`<v:rect style="width:50;height:50" stroked="true"`
	`  fillcolor="lime" strokecolor="red">`
	`  <v:stroke on="false" weight="5pt"/>`
	`</v:rect>`
	 
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

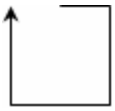
Attributes	Description

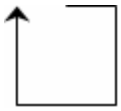
opacity (Stroke Opacity)	Specifies the amount of transparency of a stroke. Default is 1.0.
	
	[*Example*:
	
	`<v:rect style="width:50;height:50"`
	`  fillcolor="lime" strokecolor="red">`
	`  <v:stroke weight="5pt" opacity="50%"/>`
	`</v:rect>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

relid (Relationship to Part)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the relationship ID of the relationship to the image. The specified relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/image or the document shall be considered non-conformant.
	
	[*Example*: The markup specifies the associated relationship part with relationship ID `rId10` contains the corresponding relationship information:
	
	`<v:stroke … o:relid="rId10" …>`
	`</v:stroke>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

src (Stroke Image Location)	Specifies the source image to load for a stroke fill. Default is no value.
	
	[*Example*:
	
	`<v:stroke … src="myimage.gif" … >`
	`</v:stroke>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

startarrow (Line Start Arrowhead)	Specifies an arrowhead for the start of a line. Default is none. The path shall not be closed with the x command for the arrowhead to show. Allowed values are:

Attributes	Description
	• `none` • `block` • `classic` • `diamond` • `oval` • `open` [*Example*: `<v:stroke startarrow="classic"/>`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowType simple type (§19.1.3.8).
startarrowlength (Line Start Arrowhead Length)	Specifies the length of the arrowhead at the start of a line. Default is `medium`. Allowed values are: • `short` • `medium` • `long` [*Example*: `<v:stroke … startarrowlength="long" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowLength simple type (§19.1.3.7).
startarrowwidth (Line Start Arrowhead Width)	Specifies the width of the arrowhead at the start of a line. Default is `medium`. Allowed values are: • `narrow` • `medium`

Attributes	Description
	• wide [*Example*: `<v:stroke … startarrowwidth="wide" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowWidth simple type (§19.1.3.9).

title (Stroke Title)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies the title of an embedded stroke image. This is typically set to the comment property of the image, which is often blank.
	
	[*Example*:
	
	`<v:fill … o:title="alt text" … >`
	`</v:fill>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

weight (Stroke Weight)	Specifies the thickness of a stroke. Default is 1. This attribute overrides the shape's strokeweight attribute. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Stroke) is located in §A.6.1. *end note*]

19.1.2.22 textbox (Text Box)

This element is used to define text that appears inside the shape. This text can contain rich formatting and is rendered to fit inside the textboxrect defined by the path element (§19.1.2.14).

[*Example*:

```
<v:shape style="width=200;height=200" coordsize="400,400"
  fillcolor="yellow" strokecolor="maroon"
```

```
    path="m 119,0 l 148,86 238,86 166,140 192,226 119,175 46,226
    72,140 0,86 90,86 x e">
    <v:textbox inset="32pt,35pt,,">VML</v:textbox>
  </v:shape>
```

end example]

Attributes	Description

id (Unique Identifier)	Specifies a unique identifier that can be used to reference a VML object.
	
	Default is no value.
	
	[*Example*:
	
	`<v:shape … id="myShape" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

inset (Text Box Inset)	Specifies inner margin values for textbox text. Default is "0.1in, 0.05in, 0.1in, 0.05in". Missing values are set to the default. This is used if insetmode is `custom`.
	
	The internal text margin value is specified as a string containing four values, each separated by commas or spaces. The values measure inset from the left, top, right, and bottom edges of the box specified by the textboxrect attribute of the path element (§19.1.2.14).
	
	[*Example*: The text is set toward the lower right of a small square:
	
	`<v:textbox inset="20pt,30pt,10pt,10pt">`
	`VML</v:textbox>`
	
	 

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema string datatype.

insetmode (Text Inset Mode)
 	
Namespace: urn:schemas-microsoft-com:office:office
	Specifies whether the application calculates the internal text margin instead of using the inset attribute. Default is `custom`.
	
	[*Example*:
	
	`<v:textbox … o:insetmode="auto" … >`
	`</v:textbox>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_InsetMode simple type (§19.2.3.17).

singleclick (Text Box Single-Click Selection Toggle) Namespace: urn:schemas-microsoft-com:office:office	Specifies whether text is selectable with a single click. Default is `false`. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

style (Shape Styling Properties)	Specifies the CSS2 styling properties of the shape. The CSS2 (Cascading Style Sheets, Level 2) specification, a Recommendation of the World Wide Web Consortium, is available here: http://www.w3.org/TR/REC-CSS2.
	
	This attribute uses a semi-colon delimited list of "name:value" pairs, the syntax defined by section 4.1.8 of the CSS2 specification for grouped declarations (without the surrounding braces). The following tables define the allowed properties and the VML treatment of each.
	
	[*Example*:
	`<v:shape … style='position:absolute;width:100pt;height:50pt' …`
	`>`
	`</v:shape>`
	end example]
	
	Property / **Description**
	`flip` — Specifies that the orientation of a shape is flipped. Default is no value. Allowed values are:
	• x - Flip along the y-axis, reversing the x-coordinates.

Attributes	Description	
		• y - Flip along the x-axis, reversing the y-coordinates. • xy - Flip along both the y- and x-axis. • yx - Flip along both the x- and y-axis.
	`height`	Specifies the height of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`left`	Specifies the position of the left of the containing block of the shape relative to the element left of it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-bottom`	Specifies the position of the bottom of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`margin-left`	Specifies the position of the left of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are:

Attributes	Description	
		• auto - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	margin-right	Specifies the position of the right of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • auto - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	margin-top	Specifies the position of the top of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • auto - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	mso-position-horizontal	Specifies the horizontal positioning data for objects in WordprocessingML documents. Default is absolute. Allowed values are: • absolute • left • center • right • inside • outside
	mso-position-horizontal-	Specifies relative horizontal position data for objects in WordprocessingML documents. This modifies the mso-

Attributes	Description	
	relative	position-horizontal property. Default is text. Allowed values are: • margin • page • text • char
	mso-position-vertical	Specifies the vertical positioning data for objects in WordprocessingML documents. Default is absolute. Allowed values are: • absolute • top • center • bottom • inside • outside
	mso-position-vertical-relative	Specifies relative vertical position data for objects in WordprocessingML documents. This modifies the mso-position-vertical property. Default is text. Allowed values are: • margin • page • text • line
	mso-wrap-distance-bottom	Specifies the distance from the bottom of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	mso-wrap-distance-left	Specifies the distance from the left side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	mso-wrap-distance-right	Specifies the distance from the right side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	mso-wrap-	Specifies the distance from the top of the shape to the text that

Attributes	Description
	`distance-top` — wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	`mso-wrap-edited` — Specifies whether the wrap coordinates were customized by the user. If the wrap coordinates are generated by an editor, this property is `true`; otherwise they were customized by a user. Default is `false`.
	`mso-wrap-style` — Specifies the wrapping mode for text in shapes in WordprocessingML documents. Default is `square`. Allowed values are: • `square` - Wraps text inside the shape in a square. • `none` - Text does not wrap.
	`position` — Specifies the kind of positioning used to place an element. Default is `static`. When the element is contained inside a group, this property shall be `absolute`. Allowed values are: • `static` - The element is positioned according to the normal flow of the page. The `top` and `left` properties are ignored. If the object is anchored inline, this value is used. • `absolute` - The element is positioned relative to the parent, using the `top` and `left` properties. • `relative` - The element is positioned according to the normal flow of the page, but the `top` and `left` properties are used. The overlap of overlapping elements is governed by the `z-index` property.
	`rotation` — Specifies the angle that a shape is rotated, in degrees. Default is 0. Positive angles are clockwise.
	`top` — Specifies the position of the top of the containing block of the shape relative to the element above it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.

Attributes	Description
	`visibility` — Specifies whether a shape is displayed. Only `inherit` and `hidden` are used; any other values are mapped to `inherit`. Default is `inherit`. Allowed values are: • `hidden` - The shape is not visible, but is still part of the flow of the objects in the browser. Mouse events are not processed. • `inherit` - The visibility state is inherited from the parent of the shape.
	`width` — Specifies the width of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`z-index` — Specifies the display order of overlapping shapes. Default is 0. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Uses the order that the shapes appear in the page, bottom to top. • <order>- A number that represents the stacking precedence. Shapes with higher numbers are placed on top of those with lower numbers. Negative numbers are allowed.
	The following properties are only used by the textbox element (§19.1.2.22):
	Property — **Description**
	`direction` — Specifies the direction of the text in the textbox. Default is `ltr`. This property is superceded by the `mso-direction-alt` property if that is specified. Allowed values are: • `ltr` - Text is displayed left-to-right. • `rtl` - Text is displayed right-to-left.
	`layout-flow` — Determines the flow of the text layout in a textbox. Default is `horizontal`. Allowed values are:

Attributes	Description
	• horizontal - Text is displayed horizontally. • vertical - Text is displayed vertically. • vertical-ideographic - Ideographic text is displayed vertically. • horizontal-ideographic - Ideographic text is displayed horizontally.
mso-direction-alt	Specifies an alternate direction for text in textboxes. Overrides the direction property. The only allowed value is context.
mso-fit-shape-to-text	Specifies whether the shape stretches to fit the text in the textbox. Default is false.
mso-fit-text-to-shape	Specifies whether the text stretches to fit the textbox. Default is false.
mso-layout-flow-alt	Specifies the alternate layout flow for text in textboxes. This property is used instead of layout-flow when the layout flow is from bottom to top for non-ideographic languages. Its only value is bottom-to-top.
mso-next-textbox	Specifies the ID of the next textbox in a series. Used to keep track of a set of linked textboxes. Default is no value.
mso-rotate	Specifies a specific rotation value for text in a textbox. Default is 0. Allowed values are: • 0 • 90 • 180 • -90
mso-text-scale	Specifies the scaling factor for fitting text to shapes. Default is 0. This property is only used if mso-fit-text-to-shape is true.
v-text-anchor	Specifies the vertical anchoring of text in a textbox. Default is top. The alignment of a text anchor only becomes evident if mso-fit-text-to-shape is false. This property is different from the vertical-align CSS property, which is used for ideographic languages. Allowed values are: • top • middle • bottom • top-center

Attributes	Description
	• `middle-center` • `bottom-center` • `top-baseline` • `bottom-baseline` • `top-center-baseline` • `bottom-center-baseline`

The following properties are only used by the textpath element (§19.1.2.23):

Property	Description
`font`	Specifies a compound value of font settings. Default is no value. The values are the same as those of the CSS font property. The order of definitions in the string is: font-style, font-variant, font-weight, font-size, line-height, font-family.
`font-family`	Specifies the family of the font. Default is no value. The values are the same as those of the CSS font-family property.
`font-size`	Specifies the size of the font. Default is no value. The font size is defined in points. The values are the same as those of the CSS font-size property.
`font-style`	Specifies the amount of slant for a font. Default is `normal`. The values are the same as those of the CSS font-style property. Allowed values are: • `normal` • `italic` • `oblique` - Treated the same as italic.
`font-variant`	Specifies the variant style of a font. Default is `normal`. The values are the same as those of the CSS font-variant property. Allowed values are: • `normal` • `small-caps`
`font-weight`	Specifies the thickness of the letters of the font. Default is `normal`. The values are the same as those of the CSS font-weight property. Allowed values are:

Value	Description
`normal` `lighter`	Treated as non-bold.

Attributes	Description
	<table><tr><td></td><td><table><tr><td>100 200 300 400</td><td></td></tr><tr><td>bold bolder 500 600 700 800 900</td><td>Treated as bold.</td></tr></table></td></tr><tr><td>mso-text-shadow</td><td>Specifies whether a shadow is applied to the text on a text path. Default is false.</td></tr><tr><td>text-decoration</td><td>Specifies the style of text decoration. Default is none. The values are the same as those of the CSS text-decoration property. Allowed values are: • none • underline • overline • line-through • blink</td></tr><tr><td>v-rotate-letters</td><td>Specifies whether the letters of the text are rotated counterclockwise by 90 degrees. Default is false.</td></tr><tr><td>v-same-letter-heights</td><td>Specifies whether all letters are the same height regardless of initial case. If true, the lowercase letters are stretched to the height of the uppercase letters. Default is false.</td></tr><tr><td>v-text-align</td><td>Specifies the alignment of text. Default is left. Allowed values are: • left • right • center • justify • letter-justify - Distributes the extra space between</td></tr></table>

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>the letters.
• stretch-justify - Stretches the letters to fill in the space.</td></tr>
<tr><td>v-text-kern</td><td>Specifies whether kerning is turned on. Default is false.</td></tr>
<tr><td>v-text-reverse</td><td>Specifies whether the layout order of rows is reversed. Default is false. This is used for vertical text layout.</td></tr>
<tr><td>v-text-spacing-mode</td><td>Specifies the mode for letter spacing. Default is tightening. This property determines whether space is removed between each letter (tightening) or added between each letter (tracking). The amount of letter spacing change is defined by the v-text-spacing property. Allowed values are:
• tightening
• tracking</td></tr>
<tr><td>v-text-spacing</td><td>Specifies the amount of spacing for text in 100ths of single line spacing. Default is 100.</td></tr>
</table>

The line (§19.1.2.12), polyline (§19.1.2.15) and curve (§19.1.2.3) elements ignore the following properties:
• top
• left
• width
• height

The following properties are not inherited by an element that references a shapetype element (§19.1.2.20) via the id attribute:
• flip
• height
• left
• margin-left
• margin-top
• position
• rotation
• top
• visibility
• width
• z-index

The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
</table>

[*Note*: The W3C XML Schema definition of this element's content model (CT_Textbox) is located in §A.6.1. *end note*]

19.1.2.23 textpath (Text Layout Path)

This element is used to define a vector path based on the text data, font and font styles supplied. The path which results is then mapped into the region defined by the v attribute of the shape's path (§19.1.2.14).

[*Example*:

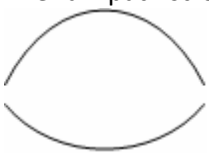
```
<v:curve from="50,100" to="400,100"
  control1="200,200" control2="300,200">
  <v:stroke color="blue"/>
  <v:fill color="yellow" color2="green" type="gradient"/>
  <v:path textpathok="true"/>
  <v:textpath on="true" style="font:normal normal normal 36pt Arial"
    fitpath="true" string="Hello, VML!"/>
</v:curve>
```

end example]

Attributes	Description

fitpath (Path Fit Toggle)	Specifies whether the text fits the path of a shape. If `true`, sizes the text to fill the path it lies out on. Default is `false`.
	
	[*Example*:
	
	`<v:textpath on="true" fitpath="true"`
	`  string="VML">`
	`</v:textpath>`
	
	  `fitpath="true"`
	  `fitpath="false"`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

fitshape (Shape Fit Toggle)	Specifies whether the text fits the bounding box of a shape. If `true`, the text is stretched out to the edges of the box that defines the entire shape. Default is `false`.
	
	[*Example*: When fitshape is `false`, the text is drawn along the first part of the path. When `true`, the text is stretched to fit the entire enclosed area of the shape.
	
	`<v:shape style="width:100;height:100"`
	`  path="m 0,500 c 250,0 750,0 1000,500 e`
	`  m 0,600 c 250,900 750,900 1000,600 e"`
	`  fillcolor="yellow" strokecolor="maroon">`
	`  <v:path textpathok="t"/>`
	`  <v:textpath on="t" fitshape="t" string="VML"/>`
	`</v:shape>`
	
	 
	
	The raw path stroke is:
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

id (Unique Identifier)	Specifies a unique identifier that can be used to reference a VML object.
	
	Default is no value.
	
	[*Example*:
	
	`<v:shape … id="myShape" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

on (Text Path Toggle)	Specifies whether the text is displayed on the textpath. Default is `false`. The textpathok attribute of the path element (§19.1.2.14) overrides this.

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>[Example:

<pre><v:line from="50,100" to="100,100">
  <v:path textpathok="false"/>
  <v:textpath on="true" string="VML"/>
</v:line></pre>
———

end example]

The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).</td></tr>
<tr><td>string (Text Path Text)</td><td>Specifies the text of the text path. Default is no value.

The possible values for this attribute are defined by the W3C XML Schema string datatype.</td></tr>
<tr><td>style (Shape Styling Properties)</td><td>Specifies the CSS2 styling properties of the shape. The CSS2 (Cascading Style Sheets, Level 2) specification, a Recommendation of the World Wide Web Consortium, is available here: http://www.w3.org/TR/REC-CSS2.

This attribute uses a semi-colon delimited list of “name:value” pairs, the syntax defined by section 4.1.8 of the CSS2 specification for grouped declarations (without the surrounding braces). The following tables define the allowed properties and the VML treatment of each.

[Example:
<pre><v:shape … style=’position:absolute;width:100pt;height:50pt’ …
>
</v:shape></pre>end example]

<table>
<tr><th>Property</th><th>Description</th></tr>
<tr><td>flip</td><td>Specifies that the orientation of a shape is flipped. Default is no value. Allowed values are:

• x - Flip along the y-axis, reversing the x-coordinates.
• y - Flip along the x-axis, reversing the y-coordinates.
• xy - Flip along both the y- and x-axis.
• yx - Flip along both the x- and y-axis.</td></tr>
<tr><td>height</td><td>Specifies the height of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are:</td></tr>
</table></td></tr>
</table>

Attributes	Description
	• auto - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
left	Specifies the position of the left of the containing block of the shape relative to the element left of it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • auto - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
margin-bottom	Specifies the position of the bottom of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • auto - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
margin-left	Specifies the position of the left of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • auto - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the

Attributes	Description	
		parent object's width.
	`margin-right`	Specifies the position of the right of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-top`	Specifies the position of the top of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`mso-position-horizontal`	Specifies the horizontal positioning data for objects in WordprocessingML documents. Default is `absolute`. Allowed values are: • `absolute` • `left` • `center` • `right` • `inside` • `outside`
	`mso-position-horizontal-relative`	Specifies relative horizontal position data for objects in WordprocessingML documents. This modifies the `mso-position-horizontal` property. Default is `text`. Allowed values are: • `margin` • `page` • `text`

Attributes	Description
	• `char`
`mso-position-vertical`	Specifies the vertical positioning data for objects in WordprocessingML documents. Default is `absolute`. Allowed values are: • `absolute` • `top` • `center` • `bottom` • `inside` • `outside`
`mso-position-vertical-relative`	Specifies relative vertical position data for objects in WordprocessingML documents. This modifies the `mso-position-vertical` property. Default is `text`. Allowed values are: • `margin` • `page` • `text` • `line`
`mso-wrap-distance-bottom`	Specifies the distance from the bottom of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
`mso-wrap-distance-left`	Specifies the distance from the left side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
`mso-wrap-distance-right`	Specifies the distance from the right side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
`mso-wrap-distance-top`	Specifies the distance from the top of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS `margin` property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
`mso-wrap-`	Specifies whether the wrap coordinates were customized by the user. If the wrap coordinates are generated by an editor, this

Attributes	Description
	`edited` — property is `true`; otherwise they were customized by a user. Default is `false`.
	`mso-wrap-style` — Specifies the wrapping mode for text in shapes in WordprocessingML documents. Default is `square`. Allowed values are: • `square` - Wraps text inside the shape in a square. • `none` - Text does not wrap.
	`position` — Specifies the kind of positioning used to place an element. Default is `static`. When the element is contained inside a group, this property shall be `absolute`. Allowed values are: • `static` - The element is positioned according to the normal flow of the page. The `top` and `left` properties are ignored. If the object is anchored inline, this value is used. • `absolute` - The element is positioned relative to the parent, using the `top` and `left` properties. • `relative` - The element is positioned according to the normal flow of the page, but the `top` and `left` properties are used. The overlap of overlapping elements is governed by the `z-index` property.
	`rotation` — Specifies the angle that a shape is rotated, in degrees. Default is 0. Positive angles are clockwise.
	`top` — Specifies the position of the top of the containing block of the shape relative to the element above it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`visibility` — Specifies whether a shape is displayed. Only `inherit` and `hidden` are used; any other values are mapped to `inherit`. Default is `inherit`. Allowed values are: • `hidden` - The shape is not visible, but is still part of the flow of the objects in the browser. Mouse events are not

Attributes	Description
	processed. • `inherit` - The visibility state is inherited from the parent of the shape.
`width`	Specifies the width of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
`z-index`	Specifies the display order of overlapping shapes. Default is 0. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Uses the order that the shapes appear in the page, bottom to top. • <order>- A number that represents the stacking precedence. Shapes with higher numbers are placed on top of those with lower numbers. Negative numbers are allowed.

The following properties are only used by the textbox element (§19.1.2.22):

Property	Description
`direction`	Specifies the direction of the text in the textbox. Default is `ltr`. This property is superceded by the `mso-direction-alt` property if that is specified. Allowed values are: • `ltr` - Text is displayed left-to-right. • `rtl` - Text is displayed right-to-left.
`layout-flow`	Determines the flow of the text layout in a textbox. Default is `horizontal`. Allowed values are: • `horizontal` - Text is displayed horizontally. • `vertical` - Text is displayed vertically. • `vertical-ideographic` - Ideographic text is displayed vertically. • `horizontal-ideographic` - Ideographic text is displayed

Attributes	Description
	horizontally.
`mso-direction-alt`	Specifies an alternate direction for text in textboxes. Overrides the `direction` property. The only allowed value is `context`.
`mso-fit-shape-to-text`	Specifies whether the shape stretches to fit the text in the textbox. Default is `false`.
`mso-fit-text-to-shape`	Specifies whether the text stretches to fit the textbox. Default is `false`.
`mso-layout-flow-alt`	Specifies the alternate layout flow for text in textboxes. This property is used instead of `layout-flow` when the layout flow is from bottom to top for non-ideographic languages. Its only value is `bottom-to-top`.
`mso-next-textbox`	Specifies the ID of the next textbox in a series. Used to keep track of a set of linked textboxes. Default is no value.
`mso-rotate`	Specifies a specific rotation value for text in a textbox. Default is 0. Allowed values are: • 0 • 90 • 180 • -90
`mso-text-scale`	Specifies the scaling factor for fitting text to shapes. Default is 0. This property is only used if `mso-fit-text-to-shape` is `true`.
`v-text-anchor`	Specifies the vertical anchoring of text in a textbox. Default is `top`. The alignment of a text anchor only becomes evident if `mso-fit-text-to-shape` is `false`. This property is different from the `vertical-align` CSS property, which is used for ideographic languages. Allowed values are: • `top` • `middle` • `bottom` • `top-center` • `middle-center` • `bottom-center` • `top-baseline` • `bottom-baseline` • `top-center-baseline` • `bottom-center-baseline`

Attributes	Description
	The following properties are only used by the textpath element (§19.1.2.23):

Property	Description
font	Specifies a compound value of font settings. Default is no value. The values are the same as those of the CSS font property. The order of definitions in the string is: font-style, font-variant, font-weight, font-size, line-height, font-family.
font-family	Specifies the family of the font. Default is no value. The values are the same as those of the CSS font-family property.
font-size	Specifies the size of the font. Default is no value. The font size is defined in points. The values are the same as those of the CSS font-size property.
font-style	Specifies the amount of slant for a font. Default is normal. The values are the same as those of the CSS font-style property. Allowed values are: • normal • italic • oblique - Treated the same as italic.
font-variant	Specifies the variant style of a font. Default is normal. The values are the same as those of the CSS font-variant property. Allowed values are: • normal • small-caps
font-weight	Specifies the thickness of the letters of the font. Default is normal. The values are the same as those of the CSS font-weight property. Allowed values are:

Value	Description
normal lighter 100 200 300 400	Treated as non-bold.
bold	Treated as bold.

Attributes	Description
	bolder 500 600 700 800 900
	`mso-text-shadow` — Specifies whether a shadow is applied to the text on a text path. Default is `false`.
	`text-decoration` — Specifies the style of text decoration. Default is `none`. The values are the same as those of the CSS text-decoration property. Allowed values are: • `none` • `underline` • `overline` • `line-through` • `blink`
	`v-rotate-letters` — Specifies whether the letters of the text are rotated counterclockwise by 90 degrees. Default is `false`.
	`v-same-letter-heights` — Specifies whether all letters are the same height regardless of initial case. If `true`, the lowercase letters are stretched to the height of the uppercase letters. Default is `false`.
	`v-text-align` — Specifies the alignment of text. Default is `left`. Allowed values are: • `left` • `right` • `center` • `justify` • `letter-justify` - Distributes the extra space between the letters. • `stretch-justify` - Stretches the letters to fill in the space.
	`v-text-kern` — Specifies whether kerning is turned on. Default is `false`.
	`v-text-reverse` — Specifies whether the layout order of rows is reversed. Default is `false`. This is used for vertical text layout.

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>v-text-spacing-mode</td><td>Specifies the mode for letter spacing. Default is tightening. This property determines whether space is removed between each letter (tightening) or added between each letter (tracking). The amount of letter spacing change is defined by the v-text-spacing property. Allowed values are:

• tightening
• tracking</td></tr>
<tr><td>v-text-spacing</td><td>Specifies the amount of spacing for text in 100ths of single line spacing. Default is 100.</td></tr>
</table>

The line (§19.1.2.12), polyline (§19.1.2.15) and curve (§19.1.2.3) elements ignore the following properties:

• top
• left
• width
• height

The following properties are not inherited by an element that references a shapetype element (§19.1.2.20) via the id attribute:

• flip
• height
• left
• margin-left
• margin-top
• position
• rotation
• top
• visibility
• width
• z-index

The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>trim (Text Path Trim Toggle)</td><td>Specifies whether extra space is removed above and below the text. If true, space reserved for ascenders and descenders is removed. Default is false.

[Example: The shape path is duplicated as a second shape and overlaid on the textpath for illustrative purposes:

<pre><v:shape style=" width:100;height:100"
  path="m 0,500 c 250,0 750,0 1000,500 e</pre>
</td></tr>
</table>

Attributes	Description

	`m 0,600 c 250,900 750,900 1000,600 e"`
	`fillcolor="yellow" strokecolor="maroon">`
	`<v:path textpathok="true"/>`
	`<v:textpath on="true" fitshape="true"`
	`string="vml" trim="true"/>`
	`</v:shape>`
	  `trim="true"`
	  `trim="false"`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

xscale (Text X-Scaling)	Specifies whether a straight text path is used instead of the shape path. If `true`, the text runs along a path from left to right along the x value of the lower boundary of the shape. Default is `false`. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextPath) is located in §A.6.1. *end note*]

19.1.3 Simple Types

The following additional simple type information in the urn:schemas-microsoft-com:vml namespace is used for documents of a transitional conformance class.

19.1.3.1 ST_EditAs (Shape Grouping Types)

This simple type specifies the different meanings of a group of shapes.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
bullseye (Bullseye Diagram)	Specifies that the group represents a bulls-eye diagram.

Enumeration Value	Description
canvas (Shape Canvas)	Specifies that the group is a regular group and does not represent a diagram.
cycle (Cycle Diagram)	Specifies that the group represents a cycle diagram.
orgchart (Organization Chart Diagram)	Specifies that the group represents an organization chart.
radial (Radial Diagram)	Specifies that the group represents a radial diagram.
stacked (Pyramid Diagram)	Specifies that the group represents a pyramid diagram.
venn (Venn Diagram)	Specifies that the group represents a Venn diagram.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_EditAs) is located in §A.6.1. *end note*]

19.1.3.2 ST_Ext (VML Extension Handling Behaviors)

This simple type specifies VML extension handling behaviors.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
backwardCompatible (Renderable)	Specifies that the VML entity may be rendered by ignoring the extension information. If edited, the extension information must be discarded.
edit (Editable)	Specifies that the VML entity may be safely rendered and edited without invalidating the extension information.
view (Not renderable)	Specifies that the VML entity is not be renderable without understanding the extension information. If the extension information cannot be understood, the downlevel image should be used to render the object.

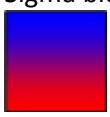
[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Ext) is located in §A.6.1. *end note*]

19.1.3.3 ST_FillMethod (Gradient Fill Computation Type)

This simple type specifies ways in which a gradient fill is computed.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
any (Application Default Fill)	Default blend

Enumeration Value	Description
	
linear (Linear Fill)	Linear blend 
linear sigma (Linear Sigma Fill)	Linear sigma blend 
none (No Gradient Fill)	No blend 
sigma (Sigma Fill)	Sigma blend 

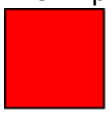
[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FillMethod) is located in §A.6.1. *end note*]

19.1.3.4 ST_FillType (Shape Fill Type)

This simple type specifies the types for fills applied to a shape.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
frame (Stretch Image to Fit)	The image is stretched to fill the shape. 
gradient (Linear Gradient)	The fill colors blend together in a linear gradient from bottom to top. 
gradientRadial (Radial Gradient)	The fill colors blend together in a radial gradient. 
pattern (Image Pattern)	The image is used to create a pattern using the fill

Enumeration Value	Description
	colors. 
solid (Solid Fill)	The fill pattern is a solid color. 
tile (Tiled Image)	The fill image is tiled. 

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FillType) is located in §A.6.1. *end note*]

19.1.3.5 ST_ImageAspect (Image Scaling Behavior)

This simple type specifies the scaling behaviors for an image applied to a stroke.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
atLeast (At Least)	Image is at least as big as imagesize. 
atMost (At Most)	Image is no bigger than imagesize. 
ignore (Ignore Aspect Ratio)	Ignore aspect issues. 

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ImageAspect) is located in §A.6.1. *end note*]

19.1.3.6 ST_ShadowType (Shadow Type)

This simple type specifies the types of shadows applied to a shape.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
double (Double Shadow)	Double shadow. color2 and offset2 are used for the second shadow's color and offset.
emboss (Embossed Shadow)	The shadow has an embossed look. Similar to double.
perspective (Perspective Shadow)	Perspective shadow.
single (Single Shadow)	Single shadow.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ShadowType) is located in §A.6.1. *end note*]

19.1.3.7 ST_StrokeArrowLength (Stroke Arrowhead Length)

This simple type specifies the lengths of a stroke arrowhead.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
long (Long Arrowhead)	Long length ➔
medium (Medium Arrowhead)	Medium length →
short (Short Arrowhead)	Short length →

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_StrokeArrowLength) is located in §A.6.1. *end note*]

19.1.3.8 ST_StrokeArrowType (Stroke Arrowhead Type)

This simple type specifies the types of arrowhead for a stroke.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
block (Block Arrowhead)	Block arrowhead ➔
classic (Classic Arrowhead)	Classic curved arrowhead ➔
diamond (Diamond Arrowhead)	Diamond arrowhead ◆

Enumeration Value	Description
none (No Arrowhead)	No arrowhead
open (Open Arrowhead)	Open arrowhead
oval (Oval Arrowhead)	Round arrowhead

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_StrokeArrowType) is located in §A.6.1. *end note*]

19.1.3.9 ST_StrokeArrowWidth (Stroke Arrowhead Width)

This simple type specifies the widths of a stroke arrowhead.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
medium (Medium Arrowhead)	Medium width
narrow (Narrow Arrowhead)	Narrow width
wide (Wide Arrowhead)	Wide width

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_StrokeArrowWidth) is located in §A.6.1. *end note*]

19.1.3.10 ST_StrokeEndCap (Stroke End Cap Type)

This simple type specifies the styles for the end of a stroke.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
flat (Flat End)	Flat end
round (Round End)	Round end

Enumeration Value	Description
	
square (Square End)	Square end 

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_StrokeEndCap) is located in §A.6.1. *end note*]

19.1.3.11 ST_StrokeJoinStyle (Line Join Type)

This simple type specifies the join styles for a polyline (§19.1.2.15).

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
bevel (Bevel Joint)	Bevel joint 
miter (Miter Joint)	Miter joint 
round (Round Joint)	Round joint 

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_StrokeJoinStyle) is located in §A.6.1. *end note*]

19.1.3.12 ST_StrokeLineStyle (Stroke Line Style)

This simple type specifies the line styles for a stroke.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
single (Single Line)	Single line 
thickBetweenThin (Thck Line Between Thin Lines)	Thick line between thin lines 
thickThin (Thick Line Outside Thin Line)	Thick line outside thin line 
thinThick (Thin Line Outside Thick Line)	Thin line outside thick line 
thinThin (Two Thin Lines)	Two thin lines 

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_StrokeLineStyle) is located in §A.6.1. *end note*]

19.2 VML - Office Drawing

It is possible to include graphical VML objects in Office Open XML documents. The elements describing the core graphical objects are defined in the VML namespace. Additional elements that describe certain advanced shape effects, metadata and relationships are defined in this namespace.

[*Note*: The VML format is a legacy format originally introduced with Office 2000 and is included and fully defined in ISO/IEC 29500 for backwards compatibility reasons. The DrawingML format is a newer and richer format created with the goal of eventually replacing any uses of VML in the Office Open XML formats. VML should be considered a transitional format included in Office Open XML for legacy reasons only and new applications that need a file format for drawings are strongly encouraged to use preferentially DrawingML .*end note*]

[*Example:* Assume the following shape exists in a document:

The basic speech bubble shape is defined using VML. The 3-D effect is defined using the extrusion element in this namespace. The specularity attribute defines the subtle sharp reflection on the edge of the shape. The color attribute sets the extrusion to a different color than the face of the shape. The rotationangle attribute sets the shape's rotation about the X- and Y-axes. The lightposition and lightposition2 attributes set the positions of the light sources that illuminate the shape.

```
<o:extrusion v:ext="view" specularity="80000f" color="#c4bc96 [2414]" on="t"
rotationangle="-5,15" lightposition="0,-50000" lightposition2="0,50000"
type="perspective"/>
```

This element is a child of the primary shape definition:

```
<v:shape id="_x0000_s1030" type="#_x0000_t62"
  style="position:absolute;left:0;text-align:left;margin-left:35.25pt;
  margin-top:60pt;width:69pt;height:57pt;z-index:251658240" adj="1675,27171"
  fillcolor="#ddd8c2 [2894]">
  <o:extrusion … />
</v:shape>
```

end example]

Throughout VML, numeric values that are allowed to take units can be specified in: cm (centimeters), mm (millimeters), in (inches), pt (points), pc (picas), px (pixels).

19.2.1 Table of Contents

This subclause is informative.

End of informative text.

19.2.2 Elements

The following elements comprise the contents of the urn:schemas-microsoft-com:office:office namespace:

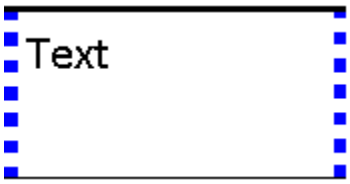
[*Note*: As the VML format is a format provided for backward compatibility, those VML elements defined in the same urn:schemas-microsoft-com:office:office namespace remain in that namespace as it is already used by millions of documents already using VML. *end note*]

19.2.2.1 bottom (Text Box Bottom Stroke)

This element specifies the stroke properties for the bottom border of a text box. It entirely supercedes its parent stroke element if its on attribute is true. Thus the default value of an unspecified attribute overrides a value specified in the parent. If the on attribute is `false` or not specified, the border is not shown.

[*Example*: The text box borders are set independently. Note that the bottom border does not inherit the weight from the parent stroke element.

```
<v:stroke weight="2.25pt">
  <o:left v:ext="view" dashstyle="1 1" color="blue" weight="5pt" on="t"/>
  <o:top v:ext="view" color="black" weight="2.25pt" on="t"/>
  <o:right v:ext="view" dashstyle="1 1" color="blue" weight="5pt" on="t"/>
  <o:bottom v:ext="view" color="black" on="t"/>
  <o:column v:ext="view" color="#f60" on="t"/>
</v:stroke>
```



end example]

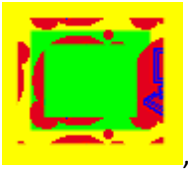
Attributes	Description

althref (Alternate Image Reference)	Specifies an alternate reference for an image in Macintosh PICT format.
	
	[*Example*:
	
	`<v:stroke … althref="myimage.pcz" … >`
	`</v:stroke>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

color (Stroke Color)	Specifies the stroke color. Overrides the strokecolor attribute of a shape. Default is `black`. See the fillcolor attribute for a list of supported named colors. [*Example*: The shape stroke is blue:

Attributes	Description

	`<v:shape … strokecolor="red" … >`
	`  <v:stroke color="blue"/>`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

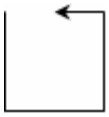
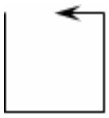
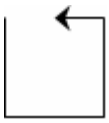
color2 (Stroke Alternate Pattern Color)	Specifies a second color for strokes, used when filltype is `pattern`. Default is no value.
	
	When a pattern fill is used for the stroke, the stroke color is used in colored parts of the source image. The color2 defines an alternate color to use in place of black in the source image.
	
	[*Example*: This unusual example is intended to demonstrate how the image and colors interact to create a patterned stroke. The yellow background shows transparency. The non-square shape and square image create an effective offset. The heavy stroke weight shows more of the image. The green shape fill shows how the stroke is overlaid on the shape.
	
	`<v:background fillcolor="yellow"/>`
	`<v:shape style="width:60;height:50"`
	`  strokecolor="red" fillcolor="lime"`
	`  path="m 0,0 l 0,1000 1000,1000 1000,0 x e">`
	`  <v:stroke filltype="pattern" weight="10pt"`
	`  src="myimage.gif" color2="blue"/>`
	`</v:shape>`
	
	  , where myimage.gif is: 
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

dashstyle (Stroke Dash Pattern)	Specifies the dot and dash pattern for a stroke. Default is `solid`. Pre-defined values are: • `solid` • `shortdash` • `shortdot` • `shortdashdot` • `shortdashdotdot` • `dot`

Attributes	Description

	• dash
	• longdash
	• dashdot
	• longdashdot
	• longdashdotdot
	
	A custom-defined dash pattern can also be specified using a series of numbers. These define the length of the dash (the drawn part of the stroke) and the length of the space between the dashes. The lengths are relative to the line width: a length of 1 is equal to the line width. The endcap style is applied to each dash but the arrow style is not. The string defines the length of the dash then the length of the space. This can be repeated to form complex dash styles. The string should always contain a pair of numbers; if it contains an odd number of numbers the last is disregarded. 0 implies a dot that is fourfold symmetrical (with round end caps, this is a circle).
	
	[*Example*:
	
	`<v:stroke dashstyle="0 2" weight="3pt"`
	`  endcap="round">`
	`</v:stroke>`
	
	 
	
	`<v:stroke dashstyle="longdashdotdot"`
	`  weight="2pt">`
	`</v:stroke>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

endarrow (Line End Arrowhead)	Specifies an arrowhead for the end of a line. Default is none. The path shall not be closed with the x command for the arrowhead to show. Allowed values are: • none • block • classic • diamond • oval • open

Attributes	Description
	[*Example*: `<v:stroke endarrow="classic"/>`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowType simple type (§19.1.3.8).
endarrowlength (Line End Arrowhead Length)	Specifies the length of the arrowhead at the end of a line. Default is `medium`. Allowed values are: • `short` • `medium` • `long` [*Example*: `<v:stroke … endarrowlength="long" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowLength simple type (§19.1.3.7).
endarrowwidth (Line End Arrowhead Width)	Specifies the width of the arrowhead at the end of a line. Default is `medium`. Allowed values are: • `narrow` • `medium` • `wide` [*Example*: `<v:stroke … endarrowwidth="wide" … />` 

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_StrokeArrowWidth simple type (§19.1.3.9).
endcap (Line End Cap)	Specifies the cap style for the end of a stroke. Default is `flat`. Allowed values are: • `flat` • `square` • `round` [*Example*: `<v:stroke … endcap="round" weight="10pt" … />`  `endcap="flat"`  `endcap="square"`  `endcap="round"` *end example*] The possible values for this attribute are defined by the ST_StrokeEndCap simple type (§19.1.3.10).
ext (VML Extension Handling Behavior) Namespace: urn:schemas-microsoft-com:vml	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).
filltype (Stroke Image Style)	Specifies the kind of fill used for the background of a stroke. Default is `solid`. Allowed values are: • `solid` - The fill pattern is solid. • `tile` - The fill image is tiled. • `pattern` - The fill image is stretched to form a pattern.

Attributes	Description

	• `frame` - The fill image becomes a border for the shape.
	
	[*Example*:
	
	`<v:shape style="width:50;height:50"`
	`  strokecolor="red"`
	`  path="m 0,0 l 0,1000 1000,1000 1000,0 x e">`
	`  <v:stroke filltype="frame" weight="10pt"`
	`  src="border.gif"/>`
	`</v:shape>`
	
	  , where border.gif is: 
	
	end example]
	
	The possible values for this attribute are defined by the ST_FillType simple type (§19.1.3.4).

forcedash (Force Dashed Outline)	Specifies whether a dashed outline is used to draw a shape when a shape has no line or fill. Default is `false`.
	
	Used by PresentationML placeholders to draw a dashed outline when there is no line and no fill for a shape.
	
	[*Example*:
	
	`<v:shape … o:forcedash="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

href (Original Image Reference)	Specifies the URL to the original image file. Used only if the picture has been linked and embedded. Default is no value.
	
	[*Example*:
	
	`<v:fill … o:href="myimage.gif" … >`
	`</v:fill>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>imagealignshape (Stoke Image Alignment)</td><td>Specifies the alignment of the stroke image. If <code>true</code>, the image is aligned with the shape. Otherwise, it is aligned with the containing scope. Default is <code>true</code>.

[Example: The top position offset shifts the image alignment relative to the containing window:

<pre><v:shape fillcolor="silver"
  style="top:20;width:50;height:50"
  path="m 0,0 l 0,1000 1000,1000 1000,0 x e">
  <v:stroke imagealignshape="false" weight="20pt"
    filltype="tile" src="myimage.gif"/>
</v:shape></pre>
  <code>imagealignshape="false"</code>

  <code>imagealignshape="false"</code>

end example]

The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).</td></tr>
<tr><td>imageaspect (Stroke Image Aspect Ratio)</td><td>Specifies how the stroke image aspect ratio is preserved. Default is <code>ignore</code>. Allowed values are:

<table><tr><th>Value</th><th>Description</th></tr><tr><td><code>ignore</code></td><td>Ignore aspect issues.</td></tr><tr><td><code>atleast</code></td><td>Image is at least as big as imagesize.</td></tr><tr><td><code>atmost</code></td><td>Image is no bigger than imagesize.</td></tr></table>
[Example:

<pre><v:stroke filltype="frame" weight="10pt"
  src="border.gif" imagealignshape="true"
  imageaspect="atleast">
</v:stroke></pre></td></tr>
</table>

Attributes	Description

	
	`imagealignshape="ignore"`
	`imagealignshape="atleast"`
	`imagealignshape="atmost"`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ImageAspect simple type (§19.1.3.5).

imagesize (Stroke Image Size)	Specifies the size of the image for the stroke. Default is the size of the image. [*Example*: `<v:stroke … imagesize="10pt,10pt" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

insetpen (Inset Border From Path)	Specifies that the border shall be displayed inside of the path defining the shape, rather than along the path (the default border placement), or outside of the path as might be done with an image.
	
	[*Example*:
	
	`<v:shape … insetpen="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

joinstyle (Line End Join Style))	Specifies the join style for line ends. Default is `round`. • `round` • `bevel` • `miter` [*Example*:

Attributes	Description

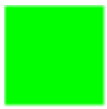
	`<v:polyline strokeweight="10pt" strokecolor="navy"`
	`  points="10pt,10pt,50pt,50pt,90pt,10pt">`
	`  <v:stroke joinstyle="bevel"/>`
	`</v:polyline>`
	
	 
	`joinstyle="round"`
	`joinstyle="bevel"`
	`joinstyle="miter"`
	
	end example]
	
	The possible values for this attribute are defined by the ST_StrokeJoinStyle simple type (§19.1.3.11).

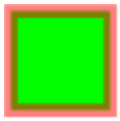
linestyle (Stroke Line Style)	Specifies the line style of the stroke. Default is `single`.
	
	• `single`
	• `thinThin`
	• `thinThick`
	• `thickThin`
	• `thickBetweenThin`
	
	[*Example*:
	
	`<v:stroke linestyle="thickThin" weight="5pt">`
	`</v:stroke>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_StrokeLineStyle simple type (§19.1.3.12).

miterlimit (Miter Joint Limit)	Specifies the smoothness of the miter joint, or the maximum distance between the inner point and outer point of a joint. This number is a multiple of the thickness of the line. Default is 8.

Attributes	Description

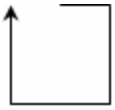
	[*Example*:
	
	`<v:stroke joinstyle="miter" weight="10pt"`
	`  miterlimit="2">`
	`</v:stroke>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema decimal datatype.

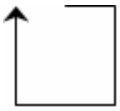
on (Stroke Toggle)	Specifies whether the stroke is displayed. Default is `true`. This attribute overrides the shape's stroke attribute.
	
	[*Example*:
	
	`<v:rect style="width:50;height:50" stroked="true"`
	`  fillcolor="lime" strokecolor="red">`
	`  <v:stroke on="false" weight="5pt"/>`
	`</v:rect>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

opacity (Stroke Opacity)	Specifies the amount of transparency of a stroke. Default is 1.0.
	
	[*Example*:
	
	`<v:rect style="width:50;height:50"`
	`  fillcolor="lime" strokecolor="red">`
	`  <v:stroke weight="5pt" opacity="50%"/>`
	`</v:rect>`
	
	 

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema string datatype.

src (Stroke Image Location)	Specifies the source image to load for a stroke fill. Default is no value.
	
	[*Example*:
	
	`<v:stroke … src="myimage.gif" … >`
	`</v:stroke>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

startarrow (Line Start Arrowhead)	Specifies an arrowhead for the start of a line. Default is `none`. The path shall not be closed with the x command for the arrowhead to show. Allowed values are: • `none` • `block` • `classic` • `diamond` • `oval` • `open` [*Example*: `<v:stroke startarrow="classic"/>`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowType simple type (§19.1.3.8).
startarrowlength (Line Start Arrowhead Length)	Specifies the length of the arrowhead at the start of a line. Default is `medium`. Allowed values are: • `short` • `medium` • `long` [*Example*:

Attributes	Description
	`<v:stroke … startarrowlength="long" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowLength simple type (§19.1.3.7).
startarrowwidth (Line Start Arrowhead Width)	Specifies the width of the arrowhead at the start of a line. Default is `medium`. Allowed values are: • `narrow` • `medium` • `wide` [*Example*: `<v:stroke … startarrowwidth="wide" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowWidth simple type (§19.1.3.9).

title (Stroke Title)	Specifies the title of an embedded stroke image. This is typically set to the comment property of the image, which is often blank.
	
	[*Example*:
	
	`<v:fill … o:title="alt text" … >`
	`</v:fill>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

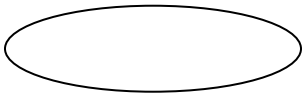
weight (Stroke Weight)	Specifies the thickness of a stroke. Default is 1. This attribute overrides the shape's strokeweight attribute. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StrokeChild) is located in §A.6.2. *end note*]

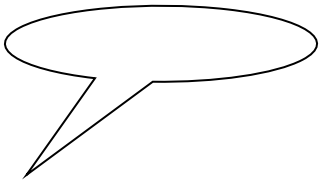
19.2.2.2 callout (Callout)

This element specifies the automatic behavior and layout parameters of callout shapes. Callout shapes are standard VML shapes that behave as callouts, providing an additional callout object which can be used to point at another location:

[*Example*: Consider the following VML shape:



If this shape is made a callout shape by adding the callout element to its shape definition, then the shape has a callout object, for example:



end example]

Attributes	Description
accentbar (Callout accent bar toggle)	Specifies whether an accent bar is used with the callout. Default is `false`. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
angle (Callout angle)	Specifies the angle that the callout makes with respect to the bounding box of the shape. Default is no value. The possible values for this attribute are defined by the ST_Angle simple type (§19.2.3.2).
distance (Callout drop distance)	Specifies the drop distance of a callout. The drop distance of a callout is measured from the edge of the shape where the pointer line starts and continues the absolute length of the distance value. If specified with no units, EMUs are assumed. Default is no value. The possible values for this attribute are defined by the W3C XML Schema string datatype.
drop (Callout drop position)	Specifies where the drop of a callout is placed. The possible values for this attribute are defined by the ST_CalloutDrop simple type (§19.2.3.4).

Attributes	Description
dropauto (Callout automatic drop toggle)	Specifies whether the callout has an automatic drop. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
ext (VML Extension Handling Behavior) Namespace: urn:schemas-microsoft-com:vml	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).
gap (Callout gap)	Specifies the distance of the callout line from the bounding rectangle of the callout. Default value is one-twelfth of an inch, in EMUs (76200). The possible values for this attribute are defined by the W3C XML Schema string datatype.
length (Callout length)	Specifies the length of the first part of a multi-segmented callout line. If specified with no units, EMUs are assumed. Default is 0. The possible values for this attribute are defined by the W3C XML Schema string datatype.
lengthspecified (Callout length toggle)	Specifies whether the length attribute is used for the callout. Default is `false`. If `true`, the length attribute is used. If `false`, a best fit is used. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
minusx (Callout flip x)	Specifies whether the callout flips to the other side of the drop tip along the x-axis when moved or resized. Default is `false`. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
minusy (Callout flip y)	Specifies whether the callout flips to the other side of the drop tip along the y-axis when moved or resized. Default is `false`. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
on (Callout toggle)	Specifies whether a shape is a callout. Default is `false`. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
textborder (Callout text border toggle)	Specifies whether a callout has a text border. Default is `true`.

Attributes	Description
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
type (Callout type)	Specifies the type of callout. Default is `rectangle`. Allowed values are: • `rectangle` • `roundedrectangle` • `oval` • `cloud` The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Callout) is located in §A.6.2. *end note*]

19.2.2.3 clippath (Shape Clipping Path)

This element specifies the path of the clipping polygon for the shape.

[*Example*:

```
<v:rect … wrapcoords="-207 -433 -207 21925 21807 21925 21807 -433 -207 -433"
o:clip="t" o:cliptowrap="t">
  <o:clippath o:v="m-207,-433r,22358l21807,21925r,-22358l-207,-433xe"/>
</v:rect>
```

end example]

Attributes	Description
v (Path Definition)	Specifies a string containing the commands that define the shape's path. This value consists of commands followed by zero or more parameters. Default is no value. The following rules apply to path strings: • Commas or spaces delimit parameters for each command. Both "m 0,0" and "m0 0" are acceptable. • A parameter that is omitted using commas is treated as having a value of zero. Thus, "c 10,10,0,0,25,13" and "c 10,10,,,25,13" are equivalent. • Parameterized paths are also allowed. In this case, the shape shall also have a formulas element (§19.1.2.6) with a list of formulas that are substituted into the path using the @ symbol followed by the number of the formula. The adj property of the shape contains the input parameters for these formulas. For example, "moveto @1@4". The evaluations of the formulas are substituted into the appropriate positions. @ also serves as a delimiter.

Attributes	Description
	The allowed commands are given below. An asterisk (*) indicates that the command is allowed to be repeated. For the qb command, the controlpoint parameter is also allowed to be repeated.

Command	Name	Parameters	Description
m	moveto	2	Start a new sub-path at the given (x,y) coordinate.
l	lineto	2*	Draw a line from the current point to the given (x,y) coordinate which becomes the new current point. Specifying a number of coordinate pairs forms a polyline.
c	curveto	6*	Draw a cubic bézier curve from the current point to the coordinate given by the final two parameters. The control points are given by the first four parameters.
x	close	0	Close the current sub-path by drawing a straight line from the current point to the original moveto point.
e	end	0	End the current set of sub-paths. A given set of sub-paths (as delimited by end) is filled. Subsequent sets of sub-paths are filled independently and superimposed on existing ones.
t	rmoveto	2*	Start a new sub-path at a coordinate relative to the current point, cp (cpx+x, cpy+y).
r	rlineto	2*	Draw a line from the current point to the given relative coordinate (cpx+x, cpy+y).
v	rcurveto	6*	Cubic bézier curve using the given coordinate relative to the current point.
nf	nofill	0	The current set of sub-paths (delimited by e) is not filled.
ns	nostroke	0	The current set of sub-paths (delimited by e) is not stroked.

Attributes	Description			
	ae	angleellipseto	6*	Draws a segment of an ellipse as described using these parameters. A straight line is drawn from the current point to the start point of the segment. The parameters are: center (x,y), size(w,h), start angle, end angle.
	al	angleellipse	6*	Same as angleellipseto except that there is an implied moveto the starting point of the segment.
	at	arcto	8*	A segment of the ellipse is drawn which starts at the angle defined by the start radius vector and ends at the angle defined by the end vector. A straight line is drawn from the current point to the start of the arc. The arc is always drawn in a counterclockwise direction. The parameters are: left, top, right, bottom, start(x,y), end(x,y). The first four values define the bounding box of an ellipse. The last four define two radial vectors.
	ar	arc	8*	Same as arcto except there is an implied moveto the start point of the arc.
	wa	clockwisearcto	8*	Same as arcto but the arc is drawn in a clockwise direction.
	wr	clockwisearc	8*	Same as arc but the arc is drawn in a clockwise direction
	qx	ellipticalqaudrantx	2*	A quarter ellipse is drawn from the current point to the given end point. The elliptical segment is initially tangential to a line parallel to the x-axis. (i.e. the segment starts out horizontal). The parameters are: end(x,y).
	qy	ellipticalquadranty	2*	Same as ellipticalquadrantx except that the elliptical segment is initially tangential to a line parallel to the y-axis (i.e. the segment starts out vertical).

Attributes	Description
	<table><tr><td>qb</td><td>quadraticbezier</td><td>2+2*</td><td>Defines one or more quadratic bézier curves by means of control points and an end point. Intermediate (on-curve) points are obtained by interpolation between successive control points as in the OpenType font specification. The sub-path need not be started in which case the sub-path is closed. In this case the last point of the sub-path defines the start point of the quadratic bézier. The parameters are: controlpoint(x,y)*, end(x,y).</td></tr></table> The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ClipPath) is located in §A.6.2. *end note*]

19.2.2.4 colormenu (UI Default Colors)

This element determines the default colors for different types of colors that can be applied to VML shapes. [*Rationale*: An application can choose to retain default colors or the last color choices a user made and present those in parts of its user interface. *end rationale*]

[*Example*:

```
<o:shapedefaults … >
  <o:colormenu v:ext="edit" fillcolor="none" extrusioncolor="#36f"/>
</o:shapedefaults>
```

end example]

Attributes	Description
ext (VML Extension Handling Behavior) Namespace: urn:schemas-microsoft-com:vml	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).
extrusioncolor	The default color associated with the 3D extrusion of a VML shape. Default is "#000000".

Attributes	Description
(Default extrusion color)	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).
fillcolor (Default fill color)	The default color associated with the fill of a VML shape. Default is "#0000FF". The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).
shadowcolor (Default shadow color)	The default color associated with the shadow of a VML shape. Default is "#80800C". The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).
strokecolor (Default stroke color)	The default color associated with the stroke of a VML shape. Default is "#FFFF00". The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorMenu) is located in §A.6.2. *end note*]

19.2.2.5 colormru (Most Recently Used Colors)

This element defines a list of up to eight colors which represent the colors most recently used by the user. [*Rationale*: An application can choose to retain the last color choices a user made, regardless of where on VML shapes they are used, and present those in parts of its user interface. *end rationale*]

[*Example*:

```
<o:shapedefaults … >
  <o:colormru v:ext="edit" colors="#a01aae,#456b69,#06f,#a1ae24,#d57811"/>
</o:shapedefaults>
```

end example]

Attributes	Description

colors (Recent colors)	A comma-separated list of up to eight most recently used colors. Default is no value. Colors should be defined using hexadecimal notation - see the ST_ColorType simple type (§20.1.2.3) for a full description.
	
	[*Example*:
	
	`<o:colormru v:ext="edit"`
	`colors="#a01aae,#456b69,#06f,#a1ae24,#d57811"/>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.
ext (VML Extension Handling Behavior) Namespace: urn:schemas-microsoft-com:vml	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).

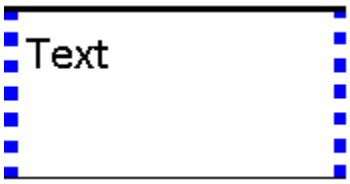
[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorMru) is located in §A.6.2. *end note*]

19.2.2.6 column (Text Box Interior Stroke)

This element specifies the stroke properties for the interior border of a text box. It entirely supercedes its parent stroke element if its on attribute is true. Thus the default value of an unspecified attribute overrides a value specified in the parent. If the on attribute is `false` or not specified, the border is not shown. [*Note*: This element is ignored if an implementation does not support multi-column text boxes. *end note*]

[*Example*: The text box borders are set independently. Note that the bottom border does not inherit the weight from the parent stroke element.

```
<v:stroke weight="2.25pt">
  <o:left v:ext="view" dashstyle="1 1" color="blue" weight="5pt" on="t"/>
  <o:top v:ext="view" color="black" weight="2.25pt" on="t"/>
  <o:right v:ext="view" dashstyle="1 1" color="blue" weight="5pt" on="t"/>
  <o:bottom v:ext="view" color="black" on="t"/>
  <o:column v:ext="view" color="#f60" on="t"/>
</v:stroke>
```



end example]

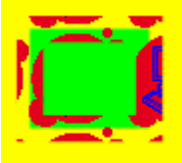
Attributes	Description
althref (Alternate Image Reference)	Specifies an alternate reference for an image in Macintosh PICT format.

Attributes	Description

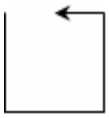
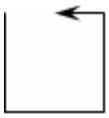
	[*Example*:
	
	`<v:stroke … althref="myimage.pcz" … >`
	`</v:stroke>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

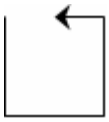
color (Stroke Color)	Specifies the stroke color. Overrides the strokecolor attribute of a shape. Default is `black`. See the fillcolor attribute for a list of supported named colors.
	
	[*Example*: The shape stroke is blue:
	
	`<v:shape … strokecolor="red" … >`
	`  <v:stroke color="blue"/>`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

color2 (Stroke Alternate Pattern Color)	Specifies a second color for strokes, used when filltype is `pattern`. Default is no value.
	
	When a pattern fill is used for the stroke, the stroke color is used in colored parts of the source image. The color2 defines an alternate color to use in place of black in the source image.
	
	[*Example*: This unusual example is intended to demonstrate how the image and colors interact to create a patterned stroke. The yellow background shows transparency. The non-square shape and square image create an effective offset. The heavy stroke weight shows more of the image. The green shape fill shows how the stroke is overlaid on the shape.
	
	`<v:background fillcolor="yellow"/>`
	`<v:shape style="width:60;height:50"`
	`  strokecolor="red" fillcolor="lime"`
	`  path="m 0,0 l 0,1000 1000,1000 1000,0 x e">`
	`  <v:stroke filltype="pattern" weight="10pt"`
	`  src="myimage.gif" color2="blue"/>`
	`</v:shape>`

Attributes	Description
	 , where myimage.gif is:  *end example*] The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

dashstyle (Stroke Dash Pattern)	Specifies the dot and dash pattern for a stroke. Default is `solid`. Pre-defined values are:
	
	• `solid`
	• `shortdash`
	• `shortdot`
	• `shortdashdot`
	• `shortdashdotdot`
	• `dot`
	• `dash`
	• `longdash`
	• `dashdot`
	• `longdashdot`
	• `longdashdotdot`
	
	A custom-defined dash pattern can also be specified using a series of numbers. These define the length of the dash (the drawn part of the stroke) and the length of the space between the dashes. The lengths are relative to the line width: a length of 1 is equal to the line width. The endcap style is applied to each dash but the arrow style is not. The string defines the length of the dash then the length of the space. This can be repeated to form complex dash styles. The string should always contain a pair of numbers; if it contains an odd number of numbers the last is disregarded. 0 implies a dot that is fourfold symmetrical (with round end caps, this is a circle).
	
	[*Example*:
	
	`<v:stroke dashstyle="0 2" weight="3pt"`
	`  endcap="round">`
	`</v:stroke>`
	
	 
	
	`<v:stroke dashstyle="longdashdotdot"`
	`  weight="2pt">`
	`</v:stroke>`

Attributes	Description
	 *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
endarrow (Line End Arrowhead)	Specifies an arrowhead for the end of a line. Default is none. The path shall not be closed with the x command for the arrowhead to show. Allowed values are: • `none` • `block` • `classic` • `diamond` • `oval` • `open` [*Example*: `<v:stroke endarrow="classic"/>`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowType simple type (§19.1.3.8).
endarrowlength (Line End Arrowhead Length)	Specifies the length of the arrowhead at the end of a line. Default is `medium`. Allowed values are: • `short` • `medium` • `long` [*Example*: `<v:stroke … endarrowlength="long" … />`  *end example*]

Attributes	Description
	The possible values for this attribute are defined by the ST_StrokeArrowLength simple type (§19.1.3.7).
endarrowwidth (Line End Arrowhead Width)	Specifies the width of the arrowhead at the end of a line. Default is `medium`. Allowed values are: • `narrow` • `medium` • `wide` [*Example*: `<v:stroke … endarrowwidth="wide" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowWidth simple type (§19.1.3.9).
endcap (Line End Cap)	Specifies the cap style for the end of a stroke. Default is `flat`. Allowed values are: • `flat` • `square` • `round` [*Example*: `<v:stroke … endcap="round" weight="10pt" … />`  `endcap="flat"`  `endcap="square"`  `endcap="round"` *end example*]

Attributes	Description
	The possible values for this attribute are defined by the ST_StrokeEndCap simple type (§19.1.3.10).
ext (VML Extension Handling Behavior) Namespace: urn:schemas-microsoft-com:vml	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).

filltype (Stroke Image Style)	Specifies the kind of fill used for the background of a stroke. Default is `solid`. Allowed values are:
	
	• `solid` - The fill pattern is solid.
	• `tile` - The fill image is tiled.
	• `pattern` - The fill image is stretched to form a pattern.
	• `frame` - The fill image becomes a border for the shape.
	
	[*Example*:
	
	`<v:shape style="width:50;height:50"`
	`  strokecolor="red"`
	`  path="m 0,0 l 0,1000 1000,1000 1000,0 x e">`
	`  <v:stroke filltype="frame" weight="10pt"`
	`  src="border.gif"/>`
	`</v:shape>`
	
	  , where border.gif is: 
	
	end example]
	
	The possible values for this attribute are defined by the ST_FillType simple type (§19.1.3.4).

forcedash (Force Dashed Outline)	Specifies whether a dashed outline is used to draw a shape when a shape has no line or fill. Default is `false`. Used by PresentationML placeholders to draw a dashed outline when there is no line and no fill for a shape. [*Example*: `<v:shape … o:forcedash="true" … >`

Attributes	Description
	`</v:shape>` *end example*] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

href (Original Image Reference)	Specifies the URL to the original image file. Used only if the picture has been linked and embedded. Default is no value.
	
	[*Example*:
	
	`<v:fill … o:href="myimage.gif" … >`
	`</v:fill>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

imagealignshape (Stoke Image Alignment)	Specifies the alignment of the stroke image. If `true`, the image is aligned with the shape. Otherwise, it is aligned with the containing scope. Default is `true`.
	
	[*Example*: The top position offset shifts the image alignment relative to the containing window:
	
	`<v:shape fillcolor="silver"`
	`style="top:20;width:50;height:50"`
	`path="m 0,0 l 0,1000 1000,1000 1000,0 x e">`
	`<v:stroke imagealignshape="false" weight="20pt"`
	`filltype="tile" src="myimage.gif"/>`
	`</v:shape>`
	
	  `imagealignshape="false"`
	
	  `imagealignshape="false"`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>imageaspect (Stroke Image Aspect Ratio)</td><td>Specifies how the stroke image aspect ratio is preserved. Default is <code>ignore</code>. Allowed values are:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><code>ignore</code></td><td>Ignore aspect issues.</td></tr>
<tr><td><code>atleast</code></td><td>Image is at least as big as imagesize.</td></tr>
<tr><td><code>atmost</code></td><td>Image is no bigger than imagesize.</td></tr>
</table>

[Example:

```
<v:stroke filltype="frame" weight="10pt"
  src="border.gif" imagealignshape="true"
  imageaspect="atleast">
</v:stroke>
```

<code>imagealignshape="ignore"</code>

<code>imagealignshape="atleast"</code>

<code>imagealignshape="atmost"</code>

end example]

The possible values for this attribute are defined by the ST_ImageAspect simple type (§19.1.3.5).</td></tr>
<tr><td>imagesize (Stroke Image Size)</td><td>Specifies the size of the image for the stroke. Default is the size of the image.

[Example:

```
<v:stroke … imagesize="10pt,10pt" … />
```

end example]

The possible values for this attribute are defined by the W3C XML Schema string datatype.</td></tr>
<tr><td>insetpen (Inset Border From Path)</td><td>Specifies that the border shall be displayed inside of the path defining the shape, rather than along the path (the default border placement), or outside of the path as might be done with an image.</td></tr>
</table>

Attributes	Description

	[*Example*:
	
	`<v:shape … insetpen="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

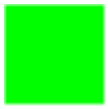
joinstyle (Line End Join Style))	Specifies the join style for line ends. Default is `round`.
	
	• `round`
	• `bevel`
	• `miter`
	
	[*Example*:
	
	`<v:polyline strokeweight="10pt" strokecolor="navy"`
	`points="10pt,10pt,50pt,50pt,90pt,10pt">`
	`<v:stroke joinstyle="bevel"/>`
	`</v:polyline>`
	
	 
	`joinstyle="round"`
	
	`joinstyle="bevel"`
	
	`joinstyle="miter"`
	
	end example]
	
	The possible values for this attribute are defined by the ST_StrokeJoinStyle simple type (§19.1.3.11).

linestyle (Stroke Line Style)	Specifies the line style of the stroke. Default is `single`. • `single` • `thinThin` • `thinThick` • `thickThin` • `thickBetweenThin`

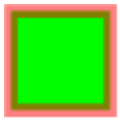
Attributes	Description

	[*Example*:
	
	`<v:stroke linestyle="thickThin" weight="5pt">`
	`</v:stroke>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_StrokeLineStyle simple type (§19.1.3.12).

miterlimit (Miter Joint Limit)	Specifies the smoothness of the miter joint, or the maximum distance between the inner point and outer point of a joint. This number is a multiple of the thickness of the line. Default is 8.
	
	[*Example*:
	
	`<v:stroke joinstyle="miter" weight="10pt"`
	`  miterlimit="2">`
	`</v:stroke>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema decimal datatype.

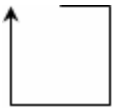
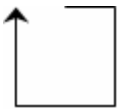
on (Stroke Toggle)	Specifies whether the stroke is displayed. Default is `true`. This attribute overrides the shape's stroke attribute.
	
	[*Example*:
	
	`<v:rect style="width:50;height:50" stroked="true"`
	`  fillcolor="lime" strokecolor="red">`
	`  <v:stroke on="false" weight="5pt"/>`
	`</v:rect>`
	
	 
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

opacity (Stroke Opacity)	Specifies the amount of transparency of a stroke. Default is 1.0.
	
	[*Example*:
	
	`<v:rect style="width:50;height:50"`
	`  fillcolor="lime" strokecolor="red">`
	`  <v:stroke weight="5pt" opacity="50%"/>`
	`</v:rect>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

src (Stroke Image Location)	Specifies the source image to load for a stroke fill. Default is no value.
	
	[*Example*:
	
	`<v:stroke … src="myimage.gif" … >`
	`</v:stroke>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

startarrow (Line Start Arrowhead)	Specifies an arrowhead for the start of a line. Default is none. The path shall not be closed with the x command for the arrowhead to show. Allowed values are: • none • block • classic • diamond • oval • open [*Example*: `<v:stroke startarrow="classic"/>`

Attributes	Description
	 *end example*] The possible values for this attribute are defined by the ST_StrokeArrowType simple type (§19.1.3.8).
startarrowlength (Line Start Arrowhead Length)	Specifies the length of the arrowhead at the start of a line. Default is `medium`. Allowed values are: • `short` • `medium` • `long` [*Example*: `<v:stroke … startarrowlength="long" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowLength simple type (§19.1.3.7).
startarrowwidth (Line Start Arrowhead Width)	Specifies the width of the arrowhead at the start of a line. Default is `medium`. Allowed values are: • `narrow` • `medium` • `wide` [*Example*: `<v:stroke … startarrowwidth="wide" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowWidth simple type (§19.1.3.9).

Attributes	Description

title (Stroke Title)	Specifies the title of an embedded stroke image. This is typically set to the comment property of the image, which is often blank.
	
	[*Example*:
	
	`<v:fill … o:title="alt text" … >`
	`</v:fill>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

weight (Stroke Weight)	Specifies the thickness of a stroke. Default is 1. This attribute overrides the shape's strokeweight attribute. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StrokeChild) is located in §A.6.2. *end note*]

19.2.2.7 complex (Complex)

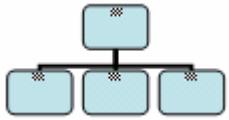
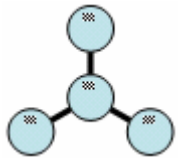
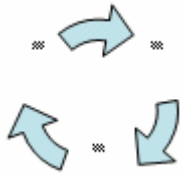
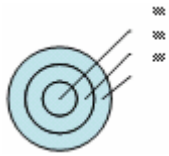
This element specifies that a shapetype contains fragments.

Attributes	Description
ext (VML Extension Handling Behavior) Namespace: urn:schemas-microsoft-com:vml	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Complex) is located in §A.6.2. *end note*]

19.2.2.8 diagram (VML Diagram)

This element specifies semantic information for a limited set of structured diagrams that have VML representations. Diagrams should be defined using DrawingML; this representation is included for compatibility with applications that rely on VML. The following diagram types have VML representations:

Diagram Type	Example (non-normative)
Organization chart	
Radial	
Cycle	
Pyramid	
Venn	
Bulls-eye	

Each of these types of diagrams contains shapes that are positioned relative to one another. Each shape also has optional associated text.

Attributes	Description

autoformat (Diagram Automatic Format)	Specifies whether the diagram is formatted automatically by the application and user overrides are locked. Default is `false`.
	
	[*Example*:
	
	`<o:diagram … autoformat="true">`
	`</o:diagram>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

Attributes	Description

autolayout (Diagram Automatic Layout)	Specifies whether the diagram elements are laid out automatically by the application and user overrides are locked. Default is `true`.
	
	[*Example*:
	
	`<o:diagram … autolayout="false">`
	`</o:diagram>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

constrainbounds (Diagram Layout Extents)	Specifies an optional, application-specific parameter related to the diagram's extents intended to be used by the application to assist laying out the diagram.
	
	[*Example*:
	
	`<o:diagram … constrainbounds="2910,2696,9773,9558">`
	`</o:diagram>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

dgmbasetextscale (Diagram Base Font Size)	Specifies the diagram's original font size. This is used in subsequent font size recalculations. If the most recent diagram font size is used to calculate the font size after a rescale, the font size would be wrong after non-isometric diagram rescalings.
	
	[*Example*:
	
	`<o:diagram … dgmbasetextscale="12">`
	`</o:diagram>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

dgmfontsize (Diagram Font Size)	Specifies the font size for attached text when a new diagram node is added.
	
	[*Example:*
	
	`<o:diagram … dgmfontsize="12">`
	`</o:diagram>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

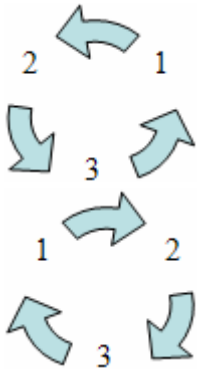
dgmscalex (Diagram Layout X Scale)	Specifies an optional, application-specific parameter related to the horizontal scaling of the diagram that is intended to be used by the application to assist laying out the diagram.
	
	[*Example*:
	
	`<o:diagram … dgmscalex="50000">`
	`</o:diagram>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

dgmscaley (Diagram Layout Y Scale)	Specifies an optional, application-specific parameter related to the vertical scaling of the diagram that is intended to be used by the application to assist laying out the diagram.
	
	[*Example*:
	
	`<o:diagram … dgmscaley="75000">`
	`</o:diagram>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

dgmstyle (Diagram Style Options)	Specifies an optional, application-specific parameter related to the styling of the diagram that is intended to be used by the application to assist in formatting the diagram.
	
	[*Example*:
	
	`<o:diagram … dgmstyle="1">`
	`</o:diagram>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema integer datatype.

ext (VML Extension Handling Behavior) Namespace: urn:schemas-	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end*

Attributes	Description
microsoft-com:vml	*rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).

reverse (Diagram Reverse Direction)	Specifies whether the order of the diagram nodes is reversed. This is only relevant to diagrams that have linear ordering.
	
	[*Example*:
	
	`<o:diagram … reverse="true">`
	`</o:diagram>`
	
	 
	
	`reverse="false"`
	`reverse="true"`
	end example
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Diagram) is located in §A.6.2. *end note*]

19.2.2.9 entry (Regroup Entry)

This element specifies a single entry in a regrouptable (§19.2.2.23). Each entry is a pair mapping a current regroupid value to an old one. This is used to restore regrouping information on the regrouped object. A value of zero indicates no previous group.

[*Example*: The zero value of the old attribute indicates that if the shapes with regroupid 1 are regrouped, the restored group was not previously grouped with any other shapes:

```
<o:regrouptable v:ext="edit">
  <o:entry new="1" old="0"/>
</o:regrouptable>
```

end example]

Attributes	Description
new (New Group ID)	Specifies the ID of the new group. Default is 0. The possible values for this attribute are defined by the W3C XML Schema int datatype.
old (Old Group ID)	Specifies the ID of the old group. Default is 0. The possible values for this attribute are defined by the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Entry) is located in §A.6.2. *end note*]

19.2.2.10 equationxml (Storage for Alternate Math Content)

This element specifies XML markup for mathematical text which can be used in place of the shape data. [*Note:* Applications are encouraged to use an open format, such as the Math format defined in ISO/IEC 29500-1, or the MathML format, a Recommendation from the World Wide Web Consortium, available at http://www.w3.org/TR/MathML/. *end note*]

[*Example*: Consider a VML object which specifies alternate math content using MathML. This object might contain the following XML markup:

```
<v:shape>
  …
  <o:equationXml contentType="mathml">
    <mrow>
      <mrow>
        <msup>
          <mi>x</mi>
          <mn>2</mn>
        </msup>
        <mo>+</mo>
        <mrow>
          <mn>4</mn>
          <mo>*</mo>
          <mi>x</mi>
        </mrow>
        <mo>+</mo>
        <mn>4</mn>
      </mrow>
      <mo>=</mo>
      <mn>0</mn>
    </mrow>
  </o:equationXml>
</v:shape>
```


The embedded MathML markup is stored within the equationxml element. *end example*]

If a producer that wants interoperability supports equations, it should use one of the following standard formats:

- Office Open XML Math (Part 1, §22.1
- W3C MathML 2.0

Attributes	Description
contentType (Content Type of Alternate Math Content)	Specifies the syntax of the markup used for the alternate math content stored in the equationxml attribute. The possible values for this attribute are defined by the ST_AlternateMathContentType simple type (§19.2.3.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_EquationXml) is located in §A.6.2. *end note*]

19.2.2.11 extrusion (3D Extrusion)

This element specifies a parallel or perspective extrusion of a 2-D shape, creating the appearance of a 3-D shape. Lighting is controlled via two independent point light sources. Extrusions are defined as either perspective or parallel.

[*Example*:

```
<v:polyline points="0pt,75pt 20pt,45pt 10pt,50pt 30pt,10pt
  50pt,50pt 40pt,45pt 60pt,75pt 0pt,75pt" fillcolor="#00a000">
  <o:extrusion on="t" backdepth="20pt"
    lightposition="30000,10000,10000"/>
</v:polyline>
```

end example]

Attributes	Description
autorotationcenter (Center of Rotation Toggle)	Specifies whether the center of rotation is the geometric center of the extrusion. Default is `false`. If `true`, the geometric center of an extruded shape is (0,0,0). If `false`, the center of rotation is determined by the rotationcenter attribute.

Attributes	Description
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

backdepth (Backward Extrusion Depth)	Specifies the amount of backward extrusion. Default is 36 pt, default units are points.
	
	[*Example*:
	
	`<o:extrusion on="true" backdepth="15pt">`
	`</o:extrusion>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

brightness (Brightness)	Specifies the overall brightness of a scene. Default is 0.3. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied (as "f" indicates the value is a fraction). For example, a value of "52429f" represents 52429/65536 or 0.8.
	
	This quantity is not specified using units. The numeric values are in the range 0 to 1 (0f to 65536f), where 0 implies darkness and 1 implies light saturation.
	
	[*Example*:
	
	`<o:extrusion on="true" brightness="0.4">`
	`</o:extrusion>`
	
	  `brightness="0"`
	  `brightness="25000f"`
	  `brightness="0.4"`
	  `brightness="0.75"`

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema string datatype.

color (Extrusion Color)	Specifies the color of the extrusion faces. This attribute is only used when colormode is `custom`. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*:
	
	`<o:extrusion on="true" color="lime"`
	`  colormode="custom">`
	`</o:extrusion>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

colormode (Extrusion Color Mode)	Specifies whether the extrusion color is defined by the color attribute or is the same as the shape's fill color. Default is `auto`.
	
	[*Example*:
	
	`<o:extrusion on="true" color="lime"`
	`  colormode="auto">`
	`</o:extrusion>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorMode simple type (§19.2.3.6).

diffusity (Diffuse Reflection)	Specifies the amount of diffusion of reflected light from an extruded shape, defined as the ratio of incident light to diffused reflected light. Default is 1. Normal values are in the range 0 to 1. This numeric value can also be specified in 1/65536-ths if a trailing "f" is

Attributes	Description

	supplied (as "f" indicates the value is a fraction). [*Example*: A value of "52429f" represents 52429/65536 or 0.8. *end example*]
	
	This quantity is not specified using units. The numeric values are in the range 0 to 1 (0f to 65536f), where 0 implies all reflected light is diffuse and 1 implies no reflected light is diffuse.
	
	Specularity and diffusity should be considered together as it is possible, though physically incorrect, to define more reflected light than incident light. This is the case if the amount of specularly reflected light and diffusely reflected light add up to more than the amount of incident light.
	
	[*Example*:
	
	`<o:extrusion on="true" diffusity=".75">`
	`</o:extrusion>`
	
	  `diffusity="0"`
	  `diffusity="0.5"`
	  `diffusity="0.75"`
	  `diffusity="1"`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

edge (Simulated Bevel)	Specifies the apparent bevel of the extrusion edges. Default is 1 point.
	
	[*Example*:
	
	`<o:extrusion on="true" edge="2pt">`
	`</o:extrusion>`

Attributes	Description
	 *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
ext (VML Extension Handling Behavior) Namespace: urn:schemas-microsoft-com:vml	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).

facet (Faceting Quality)	Specifies the quality with which the application approximates curved surfaces of an extrusion. A higher facet value produces shapes with smoother curves. A lower value reduces smoothing, resulting in curves with sharper, jagged edges. Default is 30000.
	
	Allowed values are in the range 1 to 65536, where 1 implies extremely low quality curve approximation and 65536 implies extremely high quality.
	
	[*Example*:
	
	`<o:extrusion on="true" facet="65536">`
	`</o:extrusion>`
	
	 
	`facet="65536"`
	
	 
	`facet="100"`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

foredepth (Forward Extrusion)	Specifies the amount of forward extrusion. Default is 0 pt, default units are points. [*Example*:

Attributes	Description

	`<o:extrusion on="true" foredepth="25pt">`
	`</o:extrusion>`
	 
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

lightface (Shape Face Lighting Toggle)	Specifies whether the front face of the extrusion responds to changes in the lighting. If `false`, the front face does not respond when a lighting value changes. Default is `true`.
	
	[*Example*: The front face is colored as if the shape were not extruded and lit by a 3-D light source:
	
	`<o:extrusion on="true" lightface="false">`
	`</o:extrusion>`
	 
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

lightharsh (Primary Light Harshness Toggle)	Specifies whether the primary light source is harsh. If `false`, shadow boundaries are diffused. Default is `true`.
	
	[*Example*: The secondary light source is turned off so only the primary has an effect:
	
	`<o:extrusion on="true" lightharsh="false"`
	`  lightlevel2="0">`
	`</o:extrusion>`
	  `lightharsh="false"`

Attributes	Description
	 `lightharsh="true"` *end example*] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
lightharsh2 (Secondary Light Harshness Toggle)	Specifies whether the secondary light source is harsh. If `false`, shadow boundaries defined by the secondary light source are diffused. Default is `false`. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

lightlevel (Primary Light Intensity)	Specifies the intensity of the primary light source for the scene. Default is 0.6. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied (as "f" indicates the value is a fraction). [*Example*: A value of "52429f" represents 52429/65536 or 0.8. *end example*]
	
	This quantity is not specified using units. The numeric values are in the range 0 to 1 (0f to 65536f), where 0 implies no direct light and 1 implies saturated direct light.
	
	[*Example*: The secondary light source is turned off so only the primary has an effect:
	
	`<o:extrusion on="true" lightlevel=".5"`
	`  lightlevel2="0">`
	`</o:extrusion>`
	
	  `lightlevel="1"`
	  `lightlevel="0.5"`
	  `lightlevel="0"`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

Attributes	Description
lightlevel2 (Secondary Light Intensity)	Specifies the intensity of the secondary light source for the scene. Default is 0.6. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied (as "f" indicates the value is a fraction). [*Example*: A value of "52429f" represents 52429/65536 or 0.8. *end example*] This quantity is not specified using units. The numeric values are in the range 0 to 1 (0f to 65536f), where 0 implies no direct light and 1 implies saturated direct light. The possible values for this attribute are defined by the W3C XML Schema string datatype.

lightposition (Primary Light Position)	Specifies the normalized X,Y,Z position of the primary light in a scene in 1/65536-ths. Default is "50000,0,10000". The use of a normalized vector from the shape origin effectively establishes the direction of the light relative to the shape. The distance of the light from the shape is irrelevant as the light source is treated as a directional light.
	
	The position "0,0,0" is at the center of the shape. Positive numbers move the light to the right, down and toward the viewer, respectively.
	
	[*Example*: The secondary light source is turned off so only the primary has an effect:
	
	`<o:extrusion on="true" lightlevel2="0"`
	`  lightposition="7000,-13000,20000">`
	`</o:extrusion>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

lightposition2 (Secondary Light Position)	Specifies the X,Y,Z position of the secondary light in a scene in 1/65536-ths. Default is "-50000,0,10000". The use of a normalized vector from the shape origin effectively establishes the direction of the light relative to the shape. The distance of the light from the shape is irrelevant as the light source is treated as a directional light. The position "0,0,0" is at the center of the shape. Positive numbers move the light to the right, down and toward the viewer, respectively. The possible values for this attribute are defined by the W3C XML Schema string datatype.
lockrotationcenter (Rotation Toggle)	Specifies whether the rotation of the extruded object is specified by the rotationangle attribute. If `false`, the rotation is specified by the orientation attribute. Default is `true`.

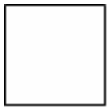
Attributes	Description

	[*Example*: The following snippets are equivalent:
	
	`<o:extrusion lockrotationcenter="false"`
	`  orientationangle="45" orientation="0,1,0">`
	`</o:extrusion>`
	
	`<o:extrusion lockrotationcenter=true`
	`  rotationangle="45"/>`
	`</o:extrusion>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

metal (Metallic Surface Toggle)	Specifies whether the surface of the extruded shape resembles metal. Default is `false`.
	
	If `true`, this attribute causes the specularly reflected light to be the material color instead of the light source color, making the object seem more metallic. To further approximate a metallic material requires that specularity be relatively high (about 1.2) and diffusity be relatively low (about 0.6).
	
	[*Example*:
	
	`<o:extrusion on="true" metal="true"`
	`  lightposition="10000,-10000,10000"`
	`  lightlevel2="0" specularity="1.2"`
	`  diffusity="0.6">`
	`</o:extrusion>`
	
	  `metal="true"`
	
	  `metal="false"`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

on (Extrusion Toggle)	Specifies whether an extrusion is displayed. Default is `false`. [*Example*:

Attributes	Description

	`<v:rect style="width=50;height=50">`
	`  <o:extrusion />`
	`</v:rect>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

orientation (Rotation Axis)	Specifies a vector in 3D space around which the shape is rotated, as given by the orientationangle attribute. Default is "100,0,0".
	
	The position "0,0,0" is at the center of the shape. Positive numbers are to the right, down and toward the viewer, respectively.
	
	[*Example*:
	
	`<o:extrusion … orientation="200,0,0">`
	`</o:extrusion>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

orientationangle (Rotation Around Axis)	Specifies the angle, in degrees, that an extrusion rotates around the orientation. Default is 0.
	
	[*Example*:
	
	`<o:extrusion … orientationangle="30">`
	`</o:extrusion>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema float datatype.

plane (Extrusion Direction)	Specifies the plane that is at right angles to the extrusion. Default is xy. Allowed values are: • xy • zx

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>

- `yz`

[*Example*:

```
<o:extrusion on="true" plane="yz"
  backdepth="100pt">
</o:extrusion>
```

end example]

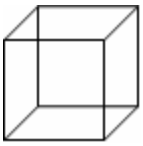
The possible values for this attribute are defined by the ST_ExtrusionPlane simple type (§19.2.3.11).
</td></tr>
<tr><td>render (Extrusion Render Mode)</td><td>

Specifies the rendering mode of the extrusion. Default is `solid`. Allowed values are:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>`solid`</td><td>Rendering displays a solid shape.</td></tr>
<tr><td>`wireframe`</td><td>Rendering displays a wireframe shape.</td></tr>
<tr><td>`boundingcube`</td><td>Rendering displays the bounding cube that contains the shape.</td></tr>
</table>

[*Example*:

```
<o:extrusion on="true" render="wireframe">
</o:extrusion>
```



end example]

The possible values for this attribute are defined by the ST_ExtrusionRender simple type (§19.2.3.12).
</td></tr>
<tr><td>rotationangle (X-Y Rotation Angle)</td><td>Specifies the rotation of the object about the x- and y-axes, in degrees. Default is "0,0". Positive angles are measured clockwise around the axis (as if viewing from the positive axis).</td></tr>
</table>

Attributes	Description

	The rotation of the object is defined by a rotation angle about the y-axis followed by the rotation angle about the x-axis. The z-axis angle is controlled by the value of the CSS style attribute's `rotation` property.
	
	[*Example*:
	
	`<o:extrusion on="t" lockrotationcenter="true"`
	`  rotationangle="10,20">`
	`</o:extrusion>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

rotationcenter (Rotation Center)	Specifies the center of rotation for a shape if autorotationcenter is `false`. The offset of the rotation is specified in terms of fractions of the shape's size. Default is "0,0,0". The position "0,0,0" is at the center of the shape. Positive numbers are to the right, down and toward the viewer, respectively. The possible values for this attribute are defined by the W3C XML Schema string datatype.
shininess (Shininess)	Specifies the concentration of the reflected light on an extrusion surface. Default is 5. The range of values should be constrained to 0-10. Reflection intensity typically grows exponentially with the shininess value. High values (8-10) approximate the shininess of a mirror and low values (2-3) approximate a speckled effect. Reflections do not mirror other objects; only pinpoint light sources are reflected. The possible values for this attribute are defined by the W3C XML Schema float datatype.

skewamt (Extrusion Skew)	Specifies the amount of skew, or length, of a parallel extrusion. Default is 50%. Applies only if the extrusion type is `parallel`. This attribute and backdepth interact to create the actual extrusion length. Allowed values are in the range 0 (0%) to 1 (100%).
	
	[*Example*:
	
	`<o:extrusion on="true" skewamt="100%">`
	`</o:extrusion>`

Attributes	Description
	 *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

skewangle (Extrusion Skew Angle)	Specifies the angle of the skew of a parallel extrusion. Default is 225 degrees. Angles are measured in degrees, counterclockwise from the negative x-axis. Applies only if the extrusion type is `parallel`.
	
	[*Example*:
	
	`<o:extrusion on="true" skewangle="25">`
	`</o:extrusion>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema float datatype.

specularity (Specularity)	Specifies the specularity of an extruded shape, defined as the ratio of incident light to specularly reflected light. Default is 0. Normal values are in the range 0 to 1. This numeric value can also be specified in 1/65536-ths if a trailing "f" is supplied (as "f" indicates the value is a fraction). [*Example*: A value of "52429f" represents 52429/65536 or 0.8. *end example*]
	
	Specularity and diffusity should be considered together as it is possible, though physically incorrect, to define more reflected light than incident light. This is the case if the amount of specularly reflected light and diffusely reflected light add up to more than the amount of incident light.
	
	[*Example*: The secondary light source is turned off so only the primary has an effect. Although the effect is subtle, the first cylinder has a sharper specular reflection on its edge:
	
	`<o:extrusion on="true" specularity="1"`
	`  lightposition="10000,-10000,10000"`
	`  lightlevel2="0">`
	`</o:extrusion>`

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>


<code>specularity="1"</code>

<code>specularity="0"</code>

end example]

The possible values for this attribute are defined by the W3C XML Schema string datatype.
</td></tr>
<tr><td>type (Extrusion Type)</td><td>
Specifies the way that the shape is extruded. Default is <code>parallel</code>. Allowed values are:

<table>
<tr><td>Value</td><td>Description</td></tr>
<tr><td><code>parallel</code></td><td>Extrusion is rendered so that the center of projection is infinitely far away; that is, the extrusion lines do not converge (unlike perspective projections).</td></tr>
<tr><td><code>perspective</code></td><td>Extrusion is rendered to a center of projection, which is the same as the vanishing point for unrotated objects.</td></tr>
</table>

[Example:

```
<o:extrusion on="true" type="parallel"
  backdepth="100pt">
</o:extrusion>
```

<code>type="parallel"</code>

<code>type="perspective"</code>

end example]

The possible values for this attribute are defined by the ST_ExtrusionType simple type
</td></tr>
</table>

Attributes	Description
	(§19.2.3.13).

viewpoint (Extrusion Viewpoint)	Specifies the viewpoint of the observer in EMUs. This is effectively the end of a vector extending from the viewpointorigin.
	
	The position "0,0,0" is at the center of the shape. Positive numbers are to the right, down and toward the viewer, respectively.
	
	[*Example*:
	
	`<o:extrusion on="true" type="perspective"`
	`  viewpoint="500000,-100000,100000">`
	`</o:extrusion>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

viewpointorigin (Extrusion Viewpoint Origin)	Specifies the origin of the viewpoint vector for perspective extrusions. This is the origin of the vector whose opposite end is given by the viewpoint attribute. This origin is always within the bounding box of the shape. Default is "0.5,-0.5".
	
	The viewpoint is specified in terms of the x and y values of the original shape. The x and y values are in the range 0.5 to -0.5 (50% to -50% of the shape's coordinate origin). Larger numbers move the viewpoint outside the bounding box.
	
	[*Example*:
	
	`<o:extrusion on="true" type="perspective"`
	`  viewpoint="500000,-100000,100000"`
	`  viewpointorigin="0,1">`
	`</o:extrusion>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Extrusion) is located in §A.6.2. *end note*]

19.2.2.12 FieldCodes (WordprocessingML Field Switches)

This element specifies the WordprocessingML field switches which shall be stored with an embedded object, using the set of field switches defined by the `LINK` field, as specified in Part 1, §17.16. This element shall only be used within a WordprocessingML document, and shall specify the exact field switches for the field which represents the object..

[*Rationale*: Legacy word processors used fields to represent embedded objects – this element stores the field switches not explicitly defined using individual Office VML Drawing elements for embeddings so as not to use the fidelity of their contents. *end rationale*]

[*Example*: The following example inserts an embedded object and specifies additional properties as defined by the LINK field.

```
<o:OLEObject …>
  <o:FieldCodes>\f 0</o:FieldCodes>
</o:OLEObject>
```

This embedded object specifies additional `LINK` field code values of `\f 0`, which specifies that the embedded object shall retain its source formatting (as defined in Part 1, §17.16).

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

19.2.2.13 fill (Shape Fill Extended Properties)

This element specifies additional properties for fills. It is used to identify additional types of gradient fills beyond those specified in the fill element (§19.1.2.5).

Attributes	Description
ext (VML Extension Handling Behavior) Namespace: urn:schemas-microsoft-com:vml	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).
type (Fill Type)	Specifies the type of fill. If specified, this overrides the value of the type attribute in the parent fill element. [*Example*: The `gradientCenter` value overrides `gradientRadial`:

Attributes	Description

	`<v:fill color2="black" focus="100%" type="gradientRadial">`
	`  <o:fill v:ext="view" type="gradientCenter"/>`
	`</v:fill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_FillType simple type (§19.2.3.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Fill) is located in §A.6.2. *end note*]

19.2.2.14 idmap (Shape ID Map)

This element specifies how shape IDs in the document have been generated. This is an optional element included to allow applications a mechanism for storing information they need to persist related to generating shape IDs.

Attributes	Description

data (Shape IDs)	Specifies the data the application uses to generate shape IDs.
	
	[*Example*: An application might choose to reserve blocks of shape ID numbers for each part in the package. Each block of 1024 shape IDs could be referred to by index and this index stored in the data attribute. The data value for a given part might then be:
	
	`<o:idmap v:ext="edit" data="1"/>`
	
	indicating that all the IDs in block 1 are reserved by this part (meaning shape IDs from 1 to 1024 cannot be used). The application's internal constraint would be that each part reserve a different set of IDs. Another part, that contains more shapes, might use:
	
	`<o:idmap v:ext="edit" data="2,3"/>`
	
	In this case, shape IDs from 1025 to 3072 [3 x 1024] cannot be used).
	
	Another implementation might choose to store more verbose information in this attribute. Yet another implementation might ignore this element completely.
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

ext (VML Extension Handling Behavior) Namespace:	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications

Attributes	Description
urn:schemas-microsoft-com:vml	that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_IdMap) is located in §A.6.2. *end note*]

19.2.2.15 ink (Ink)

This element specifies the presence of an ink object. An ink object is a VML object which allows applications to store data for ink annotations. [*Note:* Applications are encouraged to use an open-ink format, such as the Ink Markup Language (InkML). *end note*]

[*Example*:

```
<v:shape … >
  <o:ink i="…" annotation="t" contentType="application/inkml+xml"/>
</v:shape>
<v:shape … >
  <o:ink i="AMgFHQSWC+YFASAAaAwAAAAAAMA…" annotation="t"
    contentType="application/x-ms-ink"/>
</v:shape>
```

end example]

Attributes	Description

annotation (Annotation Flag)	Specifies whether the ink object was created as an annotation rather than through pen input. Default is `false`. [*Rationale* This allows an application to treat annotation ink objects as any other annotation. For example, if annotations are hidden, the application can hide the ink object. An ink object that represents primary user input through a pen can be left visible. *end rationale*]
	
	[*Example*:
	
	`<o:ink … annotation="true">`
	`</o:ink>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

contentType (Content Type)	Specifies the format of the ink content stored in the i attribute. The syntax is a content type as defined in IETF RFC 2616.

Attributes	Description
	If this attribute is omitted, the application should attempt to determine the content type by reading the contents of the i element. The possible values for this attribute are defined by the ST_ContentType simple type (§19.2.3.9).

i (Ink Data)	Specifies additional ink object information which shall be associated with the parent VML shape. The VML shape specifies the information necessary to render the ink, and this attribute can be used to store additional data about the VML shape(s) representing ink. This attribute's contents are formatted as specified by the contentType attribute, but are optional and can be ignored if not recognized.
	
	[*Example*:
	
	`<o:ink … i="…">`
	`</o:ink>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

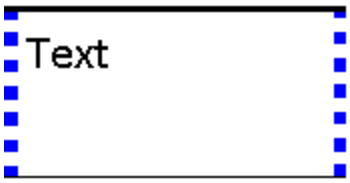
[*Note*: The W3C XML Schema definition of this element's content model (CT_Ink) is located in §A.6.2. *end note*]

19.2.2.16 left (Text Box Left Stroke)

This element specifies the stroke properties for the left border of a text box. It entirely supercedes its parent stroke element if its on attribute is true. Thus the default value of an unspecified attribute overrides a value specified in the parent. If the on attribute is `false` or not specified, the border is not shown.

[*Example*: The text box borders are set independently. The bottom border does not inherit the weight from the parent stroke element.

```
<v:stroke weight="2.25pt">
  <o:left v:ext="view" dashstyle="1 1" color="blue" weight="5pt" on="t"/>
  <o:top v:ext="view" color="black" weight="2.25pt" on="t"/>
  <o:right v:ext="view" dashstyle="1 1" color="blue" weight="5pt" on="t"/>
  <o:bottom v:ext="view" color="black" on="t"/>
  <o:column v:ext="view" color="#f60" on="t"/>
</v:stroke>
```

end example]

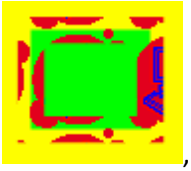
Attributes	Description

althref (Alternate Image Reference)	Specifies an alternate reference for an image in Macintosh PICT format.
	
	[*Example*:
	
	`<v:stroke … althref="myimage.pcz" … >`
	`</v:stroke>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

color (Stroke Color)	Specifies the stroke color. Overrides the strokecolor attribute of a shape. Default is `black`. See the fillcolor attribute for a list of supported named colors.
	
	[*Example*: The shape stroke is blue:
	
	`<v:shape … strokecolor="red" … >`
	`  <v:stroke color="blue"/>`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

color2 (Stroke Alternate Pattern Color)	Specifies a second color for strokes, used when filltype is `pattern`. Default is no value.
	
	When a pattern fill is used for the stroke, the stroke color is used in colored parts of the source image. The color2 defines an alternate color to use in place of black in the source image.
	
	[*Example*: This unusual example is intended to demonstrate how the image and colors interact to create a patterned stroke. The yellow background shows transparency. The non-square shape and square image create an effective offset. The heavy stroke weight shows more of the image. The green shape fill shows how the stroke is overlaid on the shape.
	
	`<v:background fillcolor="yellow"/>`
	`<v:shape style="width:60;height:50"`

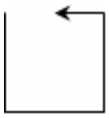
Attributes	Description

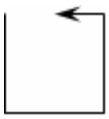
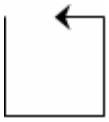
	`strokecolor="red" fillcolor="lime"`
	`path="m 0,0 l 0,1000 1000,1000 1000,0 x e">`
	`<v:stroke filltype="pattern" weight="10pt"`
	`src="myimage.gif" color2="blue"/>`
	`</v:shape>`
	
	  , where myimage.gif is: 
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

dashstyle (Stroke Dash Pattern)	Specifies the dot and dash pattern for a stroke. Default is `solid`. Pre-defined values are:
	
	• `solid`
	• `shortdash`
	• `shortdot`
	• `shortdashdot`
	• `shortdashdotdot`
	• `dot`
	• `dash`
	• `longdash`
	• `dashdot`
	• `longdashdot`
	• `longdashdotdot`
	
	A custom-defined dash pattern can also be specified using a series of numbers. These define the length of the dash (the drawn part of the stroke) and the length of the space between the dashes. The lengths are relative to the line width: a length of 1 is equal to the line width. The endcap style is applied to each dash but the arrow style is not. The string defines the length of the dash then the length of the space. This can be repeated to form complex dash styles. The string should always contain a pair of numbers; if it contains an odd number of numbers the last is disregarded. 0 implies a dot that is fourfold symmetrical (with round end caps, this is a circle).
	
	[*Example*:
	
	`<v:stroke dashstyle="0 2" weight="3pt"`
	`endcap="round">`
	`</v:stroke>`

Attributes	Description

	
	
	`<v:stroke dashstyle="longdashdotdot"`
	`  weight="2pt">`
	`</v:stroke>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

endarrow (Line End Arrowhead)	Specifies an arrowhead for the end of a line. Default is none. The path shall not be closed with the x command for the arrowhead to show. Allowed values are: • `none` • `block` • `classic` • `diamond` • `oval` • `open` [*Example*: `<v:stroke endarrow="classic"/>`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowType simple type (§19.1.3.8).
endarrowlength (Line End Arrowhead Length)	Specifies the length of the arrowhead at the end of a line. Default is `medium`. Allowed values are: • `short` • `medium` • `long` [*Example*:

Attributes	Description
	`<v:stroke … endarrowlength="long" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowLength simple type (§19.1.3.7).
endarrowwidth (Line End Arrowhead Width)	Specifies the width of the arrowhead at the end of a line. Default is `medium`. Allowed values are: • `narrow` • `medium` • `wide` [*Example*: `<v:stroke … endarrowwidth="wide" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowWidth simple type (§19.1.3.9).
endcap (Line End Cap)	Specifies the cap style for the end of a stroke. Default is `flat`. Allowed values are: • `flat` • `square` • `round` [*Example*: `<v:stroke … endcap="round" weight="10pt" … />`  `endcap="flat"`

Attributes	Description
	 `endcap="square"`  `endcap="round"` *end example*] The possible values for this attribute are defined by the ST_StrokeEndCap simple type (§19.1.3.10).
ext (VML Extension Handling Behavior) Namespace: urn:schemas-microsoft-com:vml	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).

filltype (Stroke Image Style)	Specifies the kind of fill used for the background of a stroke. Default is `solid`. Allowed values are:
	
	• `solid` - The fill pattern is solid.
	• `tile` - The fill image is tiled.
	• `pattern` - The fill image is stretched to form a pattern.
	• `frame` - The fill image becomes a border for the shape.
	
	[*Example*:
	
	`<v:shape style="width:50;height:50"`
	`  strokecolor="red"`
	`  path="m 0,0 l 0,1000 1000,1000 1000,0 x e">`
	`  <v:stroke filltype="frame" weight="10pt"`
	`  src="border.gif"/>`
	`</v:shape>`
	
	  , where border.gif is: 
	
	end example]
	
	The possible values for this attribute are defined by the ST_FillType simple type (§19.1.3.4).

Attributes	Description

forcedash (Force Dashed Outline)	Specifies whether a dashed outline is used to draw a shape when a shape has no line or fill. Default is `false`.
	
	Used by PresentationML placeholders to draw a dashed outline when there is no line and no fill for a shape.
	
	[*Example*:
	
	`<v:shape … o:forcedash="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

href (Original Image Reference)	Specifies the URL to the original image file. Used only if the picture has been linked and embedded. Default is no value.
	
	[*Example*:
	
	`<v:fill … o:href="myimage.gif" … >`
	`</v:fill>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

imagealignshape (Stoke Image Alignment)	Specifies the alignment of the stroke image. If `true`, the image is aligned with the shape. Otherwise, it is aligned with the containing scope. Default is `true`.
	
	[*Example*: The top position offset shifts the image alignment relative to the containing window:
	
	`<v:shape fillcolor="silver"`
	`  style="top:20;width:50;height:50"`
	`  path="m 0,0 l 0,1000 1000,1000 1000,0 x e">`
	`  <v:stroke imagealignshape="false" weight="20pt"`
	`  filltype="tile" src="myimage.gif"/>`
	`</v:shape>`
	
	  `imagealignshape="false"`

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>


<code>imagealignshape="false"</code>

end example]

The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
</td></tr>
<tr><td>imageaspect (Stroke Image Aspect Ratio)</td><td>
Specifies how the stroke image aspect ratio is preserved. Default is <code>ignore</code>. Allowed values are:

<table>
<tr><td>Value</td><td>Description</td></tr>
<tr><td><code>ignore</td><td>Ignore aspect issues.</td></tr>
<tr><td><code>atleast</td><td>Image is at least as big as imagesize.</td></tr>
<tr><td><code>atmost</td><td>Image is no bigger than imagesize.</td></tr>
</table>

[Example:

```
<v:stroke filltype="frame" weight="10pt"
  src="border.gif" imagealignshape="true"
  imageaspect="atleast">
</v:stroke>
```

<code>imagealignshape="ignore"</code>

<code>imagealignshape="atleast"</code>

<code>imagealignshape="atmost"</code>

end example]

The possible values for this attribute are defined by the ST_ImageAspect simple type (§19.1.3.5).
</td></tr>
<tr><td>imagesize (Stroke Image Size)</td><td>
Specifies the size of the image for the stroke. Default is the size of the image.

[Example:

```
<v:stroke … imagesize="10pt,10pt" … />
```
</td></tr>
</table>

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema string datatype.

insetpen (Inset Border From Path)	Specifies that the border shall be displayed inside of the path defining the shape, rather than along the path (the default border placement), or outside of the path as might be done with an image.
	
	[*Example*:
	
	`<v:shape … insetpen="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

joinstyle (Line End Join Style))	Specifies the join style for line ends. Default is `round`.
	
	• `round`
	• `bevel`
	• `miter`
	
	[*Example*:
	
	`<v:polyline strokeweight="10pt" strokecolor="navy"`
	`points="10pt,10pt,50pt,50pt,90pt,10pt">`
	`<v:stroke joinstyle="bevel"/>`
	`</v:polyline>`
	
	 
	`joinstyle="round"`
	`joinstyle="bevel"`
	`joinstyle="miter"`
	
	end example]

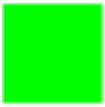
Attributes	Description
	The possible values for this attribute are defined by the ST_StrokeJoinStyle simple type (§19.1.3.11).

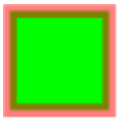
linestyle (Stroke Line Style)	Specifies the line style of the stroke. Default is `single`.
	
	• `single`
	• `thinThin`
	• `thinThick`
	• `thickThin`
	• `thickBetweenThin`
	
	[*Example*:
	
	`<v:stroke linestyle="thickThin" weight="5pt">`
	`</v:stroke>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_StrokeLineStyle simple type (§19.1.3.12).

miterlimit (Miter Joint Limit)	Specifies the smoothness of the miter joint, or the maximum distance between the inner point and outer point of a joint. This number is a multiple of the thickness of the line. Default is 8.
	
	[*Example*:
	
	`<v:stroke joinstyle="miter" weight="10pt"`
	`  miterlimit="2">`
	`</v:stroke>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema decimal datatype.

on (Stroke Toggle)	Specifies whether the stroke is displayed. Default is `true`. This attribute overrides the shape's stroke attribute. [*Example*:

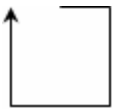
Attributes	Description

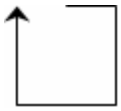
	`<v:rect style="width:50;height:50" stroked="true"`
	`  fillcolor="lime" strokecolor="red">`
	`  <v:stroke on="false" weight="5pt"/>`
	`</v:rect>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

opacity (Stroke Opacity)	Specifies the amount of transparency of a stroke. Default is 1.0.
	
	[*Example*:
	
	`<v:rect style="width:50;height:50"`
	`  fillcolor="lime" strokecolor="red">`
	`  <v:stroke weight="5pt" opacity="50%"/>`
	`</v:rect>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

src (Stroke Image Location)	Specifies the source image to load for a stroke fill. Default is no value.
	
	[*Example*:
	
	`<v:stroke … src="myimage.gif" … >`
	`</v:stroke>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

startarrow (Line Start Arrowhead)	Specifies an arrowhead for the start of a line. Default is none. The path shall not be closed with the x command for the arrowhead to show. Allowed values are: • none

Attributes	Description
	• `block` • `classic` • `diamond` • `oval` • `open` [*Example*: `<v:stroke startarrow="classic"/>`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowType simple type (§19.1.3.8).
startarrowlength (Line Start Arrowhead Length)	Specifies the length of the arrowhead at the start of a line. Default is `medium`. Allowed values are: • `short` • `medium` • `long` [*Example*: `<v:stroke … startarrowlength="long" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowLength simple type (§19.1.3.7).
startarrowwidth (Line Start Arrowhead Width)	Specifies the width of the arrowhead at the start of a line. Default is `medium`. Allowed values are: • `narrow` • `medium` • `wide` [*Example*: `<v:stroke … startarrowwidth="wide" … />`

Attributes	Description
	 *end example*] The possible values for this attribute are defined by the ST_StrokeArrowWidth simple type (§19.1.3.9).

title (Stroke Title)	Specifies the title of an embedded stroke image. This is typically set to the comment property of the image, which is often blank.
	
	[*Example*:
	
	`<v:fill … o:title="alt text" … >`
	`</v:fill>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

weight (Stroke Weight)	Specifies the thickness of a stroke. Default is 1. This attribute overrides the shape's strokeweight attribute. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StrokeChild) is located in §A.6.2. *end note*]

19.2.2.17 LinkType (Embedded Object Alternate Image Request)

This element specifies the kind of image which shall be requested from an embedded object's host application when the contents of a linked image are updated within a document. When linked images are stored in documents, the only items stored in the document are an image representation and a link to the source. This element specifies the kind of image which shall be requested from the source on update.

[*Note*: The formats available can vary based on the kind of embedded object - this information is typically queried from the embedded object's application before it is stored. This setting can be omitted, and is usually stored for performance reasons, so it is not queried on each update of the linked object. *end note*]

The possible values for this element are defined by the ST_OLELinkType simple type (§19.2.3.19).

[*Note*: The W3C XML Schema definition of this element's content model (ST_OLELinkType) is located in §A.6.2. *end note*]

19.2.2.18 lock (Shape Protections)

This element specifies locks against actions that can be effected in the UI of an authoring application or programmatically through an object model.

[*Example*: The following snippet locks the shape's aspect ratio and text from user edits.

```
<v:shape … >
  <o:lock v:ext="edit" aspectratio="t" text="t"/>
</v:shape>
```

end example]

Attributes	Description
adjusthandles (Handles Lock)	Specifies whether the handles of a shape are locked from being edited. Default is `false`. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
aspectratio (Aspect Ratio Lock)	Specifies whether the aspect ratio of a shape is locked from being edited. Default is `false`. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
cropping (Cropping Lock)	Specifies whether cropping of a shape is locked from being edited. Default is `false`. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
ext (VML Extension Handling Behavior) Namespace: urn:schemas-microsoft-com:vml	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).
grouping (Grouping Lock)	Specifies whether a shape is locked from being grouped. Default is `false`. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
position (Position Lock)	Specifies whether the position of a shape is locked from being edited. Default is `false`. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

Attributes	Description
rotation (Rotation Lock)	Specifies whether the rotation of a shape is locked from being edited. Default is `false`. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
selection (Selection Lock)	Specifies whether the shape is locked from being selectable in an editor. Default is `false`. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
shapetype (AutoShape Type Lock)	Specifies whether the AutoShape type is locked from being edited. Default is `false`. If `true`, the type of an AutoShape cannot be changed in a graphical editor. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
text (Text Lock)	Specifies whether the text attached to a shape is locked from being edited. Default is `false`. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
ungrouping (Ungrouping Lock)	Specifies whether a grouped shape is locked from being ungrouped. Default is `false`. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
verticies (Vertices Lock)	Specifies whether the vertices of a path are locked from being edited. Default is `false`. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Lock) is located in §A.6.2. *end note*]

19.2.2.19 LockedField (Embedded Object Cannot Be Refreshed)

This element specifies that the embedded object's appearance is locked - that is, that the object's current representation shall be locked to prevent any user interaction or automatic application behavior from modifying its contents.

This element shall contain no content - its presence indicates that the embedded object is locked, and its omission allows the field to be updated.

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.2.2.20 OLEObject (Embedded OLE Object)

This element specifies an embedded object.

[*Example*: The following markup defines a reference to an embedded object using Bonobo. The ProgId attribute contains the shared library that contains the widget. The content type of the referred part identifies the referenced Bonobo object.

```
<OLEObject r:id="rb1" ProgId="OAFIID:Bonobo_Sample_Calculator">
  …
</OLEObject>
```

The target of the relationship with ID `rb1`, defines the Bonobo object itself. This example shows a link to a sample Bonobo widget taken from the article http://www.ibm.com/developerworks/webservices/library/co-bnbo2.html, which also provides an introduction to Bonobo. *end example*]

[*Example*: The following demonstrates a video file embedded in a WordprocessingML document:

```
<w:object … >
  <v:shape id="_x0000_i1025" type="#_x0000_t75"
  style="width:1in;height:24pt" o:ole="">
  <v:imagedata r:id="rId4" o:title=""/>
  </v:shape>
  <o:OLEObject Type="Embed" ProgID="AVIFile" ShapeID="_x0000_i1025"
  DrawAspect="Content" ObjectID="_1219561732" r:id="rId5"/>
</w:object>
```

end example]

Attributes	Description

DrawAspect (Embedded Object Representation)	Specifies how the embedded object is represented visually in the application.
	
	[*Example*:
	
	`<o:OLEObject … DrawAspect="Content">`
	`</o:OLEObject>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_OLEDrawAspect simple type (§19.2.3.18).

id (Relationship)
 	
Namespace: …/officeDocument/2006/relationships
	Specifies the actual OLE object using a standard part relationship lookup.
	
	[*Example*:
	
	`<o:OLEObject … r:id="rId5">`
	`</o:OLEObject>`

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_RelationshipId simple type (Part 1, §22.8.2.1).

ObjectID (Unique ID for Embedded Object)	Specifies a unique ID identifying the embedded object.
	
	[*Example*: The following markup defines a reference to a linked object using KParts. The name attribute contains the shared library that contains the plugin. The item element contains the name of the plugin. The content type of the referred part would identifies the referenced KParts object.
	
	`<oleLink r:id="rKp1" progId="libhtmlvalidatorplugin">`
	`…`
	`</oleLink>`
	
	The following XML, contained in the target of the relationship with ID `rKp1`, defines the KPart object, and follows the kpartgui DTD:
	
	`<!DOCTYPE kpartgui SYSTEM "kpartgui.dtd">`
	`<kpartgui library="libhtmlvalidatorplugin"`
	`name="htmlvalidatorplugin" version="1" >`
	`<MenuBar>`
	`<Menu name="tools"><Text>&Tools</Text>`
	`<Action name="validatewebpage"/>`
	`</Menu>`
	`</MenuBar>`
	`</kpartgui>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

ProgID (Object Link Identifier)	Specifies the embedded object server application associated with the embedded object.
	
	[*Example*:
	
	`<o:OLEObject … ProgID="AVIFile">`
	`</o:OLEObject>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

ShapeID (Embedded Object	Specifies the shape with which the embedded object is associated. A VML shape provides the visual placeholder for an embedded object and this attribute is set to the id

Attributes	Description

Shape)	of the placeholder shape.
	
	[*Example*:
	
	`<o:OLEObject … ShapeID="_x0000_i1025">`
	`</o:OLEObject>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

Type (Embedded Object Type)	Specifies the kind of embedded object connection.
	
	[*Example*:
	
	`<o:OLEObject … Type="Embed">`
	`</o:OLEObject>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_OLEType simple type (§19.2.3.20).

UpdateMode (Update Mode for Embedded Object)	Specifies how the object is updated with new data if the Type is `Link` - automatically or on-demand by the user.
	
	[*Example*:
	
	`<o:OLEObject … UpdateMode="Always">`
	`</o:OLEObject>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_OLEUpdateMode simple type (§19.2.3.21).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OLEObject) is located in §A.6.2. *end note*]

19.2.2.21 proxy (Shape Reference)

This element specifies an entry in a r element rule that contains a reference to one or more shapes that are participating in the rule.

[*Example*: The following rule defines a connection between two shapes. The shape with id _s1036 connects shape _s1033 to _s1032:

```
<o:shapelayout v:ext="edit">
  <o:rules v:ext="edit">
    <o:r id="V:Rule1" type="connector" idref="#_s1036">
      <o:proxy start="" idref="#_s1033" connectloc="0"/>
      <o:proxy end="" idref="#_s1032" connectloc="2"/>
    </o:r>
  </o:rules>
</o:shapelayout>
```

end example]

Attributes	Description
connectloc (Connection Location)	Specifies the location on the shape where the connector is attached. The value is an index into the list of connection points defined in the shape - see the connectlocs attribute. Default is 0. Only used in a connector rule. The possible values for this attribute are defined by the W3C XML Schema int datatype.
end (End Point Connection Flag)	Specifies whether the connector's end point is connected to the shape. Default is `false`. Only used in a connector rule. The possible values for this attribute are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).
idref (Proxy Shape Reference)	Specifies a reference to a shape in the current document. Default is no value. A shape name is used as the reference mechanism; this is not a relationship ID. This attribute indicates that the referenced shape is part of this rule. Two or more proxy elements are used for an alignment rule. A connector rule uses one or two, indicating which shapes the connector is attached to. The possible values for this attribute are defined by the W3C XML Schema string datatype.
start (Start Point Connection Flag)	Specifies whether the connector's start point is connected to the shape. Default is `false`. Only used in a connector rule. If both start and end are specified the later one takes precedence. The possible values for this attribute are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Proxy) is located in §A.6.2. *end note*]

19.2.2.22 r (Rule)

This element specifies a rule entry in a rules element rule set that describes how a certain shape or set of shapes behaves during editing.

[*Example*: The following rule defines a connection between two shapes. The shape with id _s1036 connects shape _s1033 to _s1032:

```
<o:shapelayout v:ext="edit">
  <o:rules v:ext="edit">
    <o:r id="V:Rule1" type="connector" idref="#_s1036">
      <o:proxy start="" idref="#_s1033" connectloc="0"/>
      <o:proxy end="" idref="#_s1032" connectloc="2"/>
    </o:r>
  </o:rules>
</o:shapelayout>
```

end example]

Attributes	Description
how (Alignment Rule Type)	Specifies the kind of alignment for an alignment rule. Default is no value. Allowed values are: • top • middle • bottom • left • center • right The possible values for this attribute are defined by the ST_How simple type (§19.2.3.15).
id (Rule ID)	Specifies an identifier for the rule. Default is no value. The possible values for this attribute are defined by the W3C XML Schema string datatype.
idref (Rule Shape Reference)	Specifies a reference to a shape in the current document that is the primary shape in the rule. [*Example*: For a connector rule, the connector. *end example*] Default is no value. A shape name is used as the reference mechanism; this is not a relationship ID. The possible values for this attribute are defined by the W3C XML Schema string datatype.
type (Rule Type)	Specifies the kind of the rule. Default is no value. Allowed values are: • arc • callout • connector • align

Attributes	Description
	The possible values for this attribute are defined by the ST_RType simple type (§19.2.3.22).

[*Note*: The W3C XML Schema definition of this element's content model (CT_R) is located in §A.6.2. *end note*]

19.2.2.23 regrouptable (Shape Grouping History)

This element specifies a list of entries which describe how shapes were previously grouped so they can be regrouped. The regroupid attribute of shapes indicates which shapes belong together when a regroup is performed. The regrouptable tracks the previous regroupid that should be assigned to all shapes with the given current regroupid.

[*Example*: Consider a document containing two rectangles and a circle. The rectangles are grouped together, then that group is grouped with the circle. This new group is then ungrouped, leaving the circle and grouped rectangles. The document might contain the following snippets:

```
<v:oval … o:regroupid="1"/>
<v:group … o:regroupid="1"/>
  <v:rect … />
  <v:rect … />
</v:group>

<o:regrouptable v:ext="edit">
  <o:entry new="1" old="0"/>
</o:regrouptable>
```

The regroupid attribute indicates that the shapes with regroupid 1 were previously grouped together. The entry indicates that if those shapes are regrouped, the new group formed should not have a regroupid value as it was not previously ungrouped.

If the two rectangles are ungrouped, the document reflects that the rectangles were previously grouped and that their old group was previously grouped:

```
<v:oval … o:regroupid="1"/>
<v:rect … o:regroupid="2"/>
<v:rect … o:regroupid="2"/>

<o:regrouptable v:ext="edit">
  <o:entry new="1" old="0"/>
  <o:entry new="2" old="1"/>
</o:regrouptable>
```

end example]

Attributes	Description
ext (VML Extension Handling Behavior) Namespace: urn:schemas-microsoft-com:vml	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RegroupTable) is located in §A.6.2. *end note*]

19.2.2.24 rel (Diagram Relationship)

This element specifies a relationship between two diagram nodes. An optional third node that exists between the primary two can also be included. The relationship has an implicit order since it describes the source and destination nodes.

[*Example*: In the cycle diagram below, shape 1036 (the shape that is the text box for the text "2") is the first node. A relationship exists between shape 1036 and shape 1044 (the text box containing "1"). In between those shapes is shape 1038 (the yellow arrow).

```
<o:relationtable v:ext="edit">
  <o:rel v:ext="edit" idsrc="#_s1036" iddest="#_s1036"/>
  <o:rel v:ext="edit" idsrc="#_s1042" iddest="#_s1036" idcntr="#_s1043"/>
  <o:rel v:ext="edit" idsrc="#_s1044" iddest="#_s1042" idcntr="#_s1045"/>
  <o:rel v:ext="edit" idsrc="#_s1036" iddest="#_s1044" idcntr="#_s1038"/>
</o:relationtable>

<v:rect id="_s1036" … >
  <v:textbox … ><…>2</…></v:textbox>
</v:rect>

<v:rect id="_s1044" … >
  <v:textbox … ><…>1</…></v:textbox>
</v:rect>

<v:shape id="_s1038" … />
```

end example]

Attributes	Description
ext (VML Extension Handling Behavior) Namespace: urn:schemas-microsoft-com:vml	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).

idcntr (Diagram Relationship Center Shape)	Specifies the optional identifier of the shape that exists between the source and destination shapes. This is omitted if the relationship does not have a shape between the source and destination shapes.
	
	[*Example*:
	
	`<o:rel … idcntr="#s_1038">`
	`</o:rel>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

iddest (Diagram Relationship Destination Shape)	Specifies the identifier of the shape at the destination of the relationship.
	
	[*Example*:
	
	`<o:rel … iddest="#s_1044">`
	`</o:rel>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

idsrc (Diagram Relationship Source Shape)	Specifies the identifier of the shape at the source of the relationship.
	
	[*Example*:
	
	`<o:rel … idsrc="#s_1036">`
	`</o:rel>`

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Relation) is located in §A.6.2. *end note*]

19.2.2.25 relationtable (Diagram Relationship Table)

This element specifies a list that describes the relationships among diagram nodes.

[*Example*: The following table describes the parent-child relationships for shapes in an organization chart. The first entry describes the top-level shape in the diagram. The next two rows describe that the shapes are subordinates to the first shape. Shape 1029 is a subordinate of shape 1028. Shape 1032, a connector in this case, is in between the two.

```
<o:relationtable v:ext="edit">
  <o:rel v:ext="edit" idsrc="#_s1028" iddest="#_s1028"/>
  <o:rel v:ext="edit" idsrc="#_s1029" iddest="#_s1028" idcntr="#_s1032"/>
  <o:rel v:ext="edit" idsrc="#_s1030" iddest="#_s1028" idcntr="#_s1033"/>
</o:relationtable>
```

end example]

Attributes	Description
ext (VML Extension Handling Behavior) Namespace: urn:schemas-microsoft-com:vml	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).

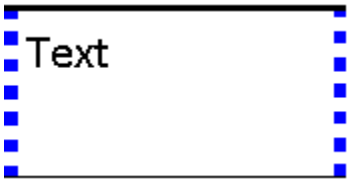
[*Note*: The W3C XML Schema definition of this element's content model (CT_RelationTable) is located in §A.6.2. *end note*]

19.2.2.26 right (Text Box Right Stroke)

This element specifies the stroke properties for the right border of a text box. It entirely supercedes its parent stroke element if its on attribute is true. Thus the default value of an unspecified attribute overrides a value specified in the parent. If the on attribute is `false` or not specified, the border is not shown.

[*Example*: The text box borders are set independently. The bottom border does not inherit the weight from the parent stroke element.

```
<v:stroke weight="2.25pt">
  <o:left v:ext="view" dashstyle="1 1" color="blue" weight="5pt" on="t"/>
  <o:top v:ext="view" color="black" weight="2.25pt" on="t"/>
  <o:right v:ext="view" dashstyle="1 1" color="blue" weight="5pt" on="t"/>
  <o:bottom v:ext="view" color="black" on="t"/>
  <o:column v:ext="view" color="#f60" on="t"/>
</v:stroke>
```



end example]

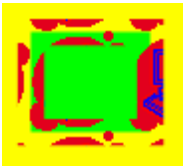
Attributes	Description

althref (Alternate Image Reference)	Specifies an alternate reference for an image in Macintosh PICT format.
	
	[*Example*:
	
	`<v:stroke … althref="myimage.pcz" … >`
	`</v:stroke>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

color (Stroke Color)	Specifies the stroke color. Overrides the strokecolor attribute of a shape. Default is `black`. See the fillcolor attribute for a list of supported named colors.
	
	[*Example*: The shape stroke is blue:
	
	`<v:shape … strokecolor="red" … >`
	`  <v:stroke color="blue"/>`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

color2 (Stroke Alternate Pattern	Specifies a second color for strokes, used when filltype is `pattern`. Default is no value.

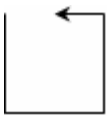
Attributes	Description

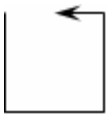
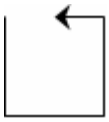
Color)	When a pattern fill is used for the stroke, the stroke color is used in colored parts of the source image. The color2 defines an alternate color to use in place of black in the source image.
	
	[*Example*: This unusual example is intended to demonstrate how the image and colors interact to create a patterned stroke. The yellow background shows transparency. The non-square shape and square image create an effective offset. The heavy stroke weight shows more of the image. The green shape fill shows how the stroke is overlaid on the shape.
	
	`<v:background fillcolor="yellow"/>`
	`<v:shape style="width:60;height:50"`
	`  strokecolor="red" fillcolor="lime"`
	`  path="m 0,0 l 0,1000 1000,1000 1000,0 x e">`
	`  <v:stroke filltype="pattern" weight="10pt"`
	`  src="myimage.gif" color2="blue"/>`
	`</v:shape>`
	
	  , where myimage.gif is: 
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

dashstyle (Stroke Dash Pattern)	Specifies the dot and dash pattern for a stroke. Default is `solid`. Pre-defined values are: • `solid` • `shortdash` • `shortdot` • `shortdashdot` • `shortdashdotdot` • `dot` • `dash` • `longdash` • `dashdot` • `longdashdot` • `longdashdotdot` A custom-defined dash pattern can also be specified using a series of numbers. These define the length of the dash (the drawn part of the stroke) and the length of the space between the dashes. The lengths are relative to the line width: a length of 1 is equal to the line width. The endcap style is applied to each dash but the arrow style is not. The string defines the length of the dash then the length of the space. This can be repeated

Attributes	Description

	to form complex dash styles. The string should always contain a pair of numbers; if it contains an odd number of numbers the last is disregarded. 0 implies a dot that is fourfold symmetrical (with round end caps, this is a circle).
	
	[*Example*:
	
	`<v:stroke dashstyle="0 2" weight="3pt"`
	`  endcap="round">`
	`</v:stroke>`
	
	 
	
	`<v:stroke dashstyle="longdashdotdot"`
	`  weight="2pt">`
	`</v:stroke>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

endarrow (Line End Arrowhead)	Specifies an arrowhead for the end of a line. Default is none. The path shall not be closed with the x command for the arrowhead to show. Allowed values are: • `none` • `block` • `classic` • `diamond` • `oval` • `open` [*Example*: `<v:stroke endarrow="classic"/>`  *end example*]

Attributes	Description
	The possible values for this attribute are defined by the ST_StrokeArrowType simple type (§19.1.3.8).
endarrowlength (Line End Arrowhead Length)	Specifies the length of the arrowhead at the end of a line. Default is `medium`. Allowed values are: • `short` • `medium` • `long` [*Example*: `<v:stroke … endarrowlength="long" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowLength simple type (§19.1.3.7).
endarrowwidth (Line End Arrowhead Width)	Specifies the width of the arrowhead at the end of a line. Default is `medium`. Allowed values are: • `narrow` • `medium` • `wide` [*Example*: `<v:stroke … endarrowwidth="wide" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowWidth simple type (§19.1.3.9).
endcap (Line End Cap)	Specifies the cap style for the end of a stroke. Default is `flat`. Allowed values are: • `flat` • `square` • `round`

Attributes	Description
	[*Example*: `<v:stroke … endcap="round" weight="10pt" … />`  `endcap="flat"`  `endcap="square"`  `endcap="round"` *end example*] The possible values for this attribute are defined by the ST_StrokeEndCap simple type (§19.1.3.10).
ext (VML Extension Handling Behavior) Namespace: urn:schemas-microsoft-com:vml	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).

filltype (Stroke Image Style)	Specifies the kind of fill used for the background of a stroke. Default is `solid`. Allowed values are:
	
	• `solid` - The fill pattern is solid.
	• `tile` - The fill image is tiled.
	• `pattern` - The fill image is stretched to form a pattern.
	• `frame` - The fill image becomes a border for the shape.
	
	[*Example*:
	
	`<v:shape style="width:50;height:50"`
	`  strokecolor="red"`
	`  path="m 0,0 l 0,1000 1000,1000 1000,0 x e">`
	`  <v:stroke filltype="frame" weight="10pt"`
	`  src="border.gif"/>`
	`</v:shape>`

Attributes	Description
	 , where border.gif is:  *end example*] The possible values for this attribute are defined by the ST_FillType simple type (§19.1.3.4).

forcedash (Force Dashed Outline)	Specifies whether a dashed outline is used to draw a shape when a shape has no line or fill. Default is `false`.
	
	Used by PresentationML placeholders to draw a dashed outline when there is no line and no fill for a shape.
	
	[*Example*:
	
	`<v:shape … o:forcedash="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

href (Original Image Reference)	Specifies the URL to the original image file. Used only if the picture has been linked and embedded. Default is no value.
	
	[*Example*:
	
	`<v:fill … o:href="myimage.gif" … >`
	`</v:fill>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

imagealignshape (Stoke Image Alignment)	Specifies the alignment of the stroke image. If `true`, the image is aligned with the shape. Otherwise, it is aligned with the containing scope. Default is `true`.
	
	[*Example*: The top position offset shifts the image alignment relative to the containing window:
	
	`<v:shape fillcolor="silver"`
	`style="top:20;width:50;height:50"`
	`path="m 0,0 l 0,1000 1000,1000 1000,0 x e">`

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>

```
  <v:stroke imagealignshape="false" weight="20pt"
    filltype="tile" src="myimage.gif"/>
</v:shape>
```

`imagealignshape="false"`

`imagealignshape="false"`

end example]

The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
</td></tr>
<tr><td>imageaspect (Stroke Image Aspect Ratio)</td><td>

Specifies how the stroke image aspect ratio is preserved. Default is `ignore`. Allowed values are:

<table>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>`ignore`</td><td>Ignore aspect issues.</td></tr>
<tr><td>`atleast`</td><td>Image is at least as big as imagesize.</td></tr>
<tr><td>`atmost`</td><td>Image is no bigger than imagesize.</td></tr>
</table>

[*Example*:

```
<v:stroke filltype="frame" weight="10pt"
  src="border.gif" imagealignshape="true"
  imageaspect="atleast">
</v:stroke>
```

`imagealignshape="ignore"`

`imagealignshape="atleast"`

`imagealignshape="atmost"`
</td></tr>
</table>

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_ImageAspect simple type (§19.1.3.5).
imagesize (Stroke Image Size)	Specifies the size of the image for the stroke. Default is the size of the image. [*Example*: `<v:stroke … imagesize="10pt,10pt" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

insetpen (Inset Border From Path)	Specifies that the border shall be displayed inside of the path defining the shape, rather than along the path (the default border placement), or outside of the path as might be done with an image.
	
	[*Example*:
	
	`<v:shape … insetpen="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

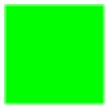
joinstyle (Line End Join Style))	Specifies the join style for line ends. Default is `round`.
	
	• `round`
	• `bevel`
	• `miter`
	
	[*Example*:
	
	`<v:polyline strokeweight="10pt" strokecolor="navy"`
	`points="10pt,10pt,50pt,50pt,90pt,10pt">`
	`<v:stroke joinstyle="bevel"/>`
	`</v:polyline>`
	
	  `joinstyle="round"`

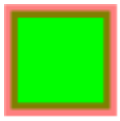
Attributes	Description
	 `joinstyle="bevel"`  `joinstyle="miter"` *end example*] The possible values for this attribute are defined by the ST_StrokeJoinStyle simple type (§19.1.3.11).

linestyle (Stroke Line Style)	Specifies the line style of the stroke. Default is `single`.
	
	• `single`
	• `thinThin`
	• `thinThick`
	• `thickThin`
	• `thickBetweenThin`
	
	[*Example*:
	
	`<v:stroke linestyle="thickThin" weight="5pt">`
	`</v:stroke>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_StrokeLineStyle simple type (§19.1.3.12).

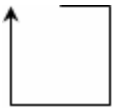
miterlimit (Miter Joint Limit)	Specifies the smoothness of the miter joint, or the maximum distance between the inner point and outer point of a joint. This number is a multiple of the thickness of the line. Default is 8.
	
	[*Example*:
	
	`<v:stroke joinstyle="miter" weight="10pt"`
	`  miterlimit="2">`
	`</v:stroke>`
	
	 

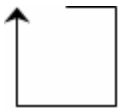
Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema decimal datatype.

on (Stroke Toggle)	Specifies whether the stroke is displayed. Default is `true`. This attribute overrides the shape's stroke attribute.
	
	[*Example*:
	
	`<v:rect style="width:50;height:50" stroked="true"`
	`  fillcolor="lime" strokecolor="red">`
	`  <v:stroke on="false" weight="5pt"/>`
	`</v:rect>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

opacity (Stroke Opacity)	Specifies the amount of transparency of a stroke. Default is 1.0.
	
	[*Example*:
	
	`<v:rect style="width:50;height:50"`
	`  fillcolor="lime" strokecolor="red">`
	`  <v:stroke weight="5pt" opacity="50%"/>`
	`</v:rect>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

src (Stroke Image Location)	Specifies the source image to load for a stroke fill. Default is no value. [*Example*: `<v:stroke … src="myimage.gif" … >`

Attributes	Description
	`</v:stroke>` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
startarrow (Line Start Arrowhead)	Specifies an arrowhead for the start of a line. Default is `none`. The path shall not be closed with the x command for the arrowhead to show. Allowed values are: • `none` • `block` • `classic` • `diamond` • `oval` • `open` [*Example*: `<v:stroke startarrow="classic"/>`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowType simple type (§19.1.3.8).
startarrowlength (Line Start Arrowhead Length)	Specifies the length of the arrowhead at the start of a line. Default is `medium`. Allowed values are: • `short` • `medium` • `long` [*Example*: `<v:stroke … startarrowlength="long" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowLength simple

Attributes	Description
	type (§19.1.3.7).
startarrowwidth (Line Start Arrowhead Width)	Specifies the width of the arrowhead at the start of a line. Default is `medium`. Allowed values are: • `narrow` • `medium` • `wide` [*Example*: `<v:stroke … startarrowwidth="wide" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowWidth simple type (§19.1.3.9).

title (Stroke Title)	Specifies the title of an embedded stroke image. This is typically set to the comment property of the image, which is often blank.
	
	[*Example*:
	
	`<v:fill … o:title="alt text" … >`
	`</v:fill>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

weight (Stroke Weight)	Specifies the thickness of a stroke. Default is 1. This attribute overrides the shape's strokeweight attribute. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StrokeChild) is located in §A.6.2. *end note*]

19.2.2.27 rules (Rule Set)

This element specifies a list of rule entries which describe how a certain shape or sets of shapes should behave during editing.

[*Example*: The following rule defines a connection between two shapes. The shape with id _s1036 connects shape _s1033 to _s1032:

```
<o:shapelayout v:ext="edit">
  <o:rules v:ext="edit">
    <o:r id="V:Rule1" type="connector" idref="#_s1036">
      <o:proxy start="" idref="#_s1033" connectloc="0"/>
      <o:proxy end="" idref="#_s1032" connectloc="2"/>
    </o:r>
  </o:rules>
</o:shapelayout>
```

end example]

Attributes	Description
ext (VML Extension Handling Behavior) Namespace: urn:schemas-microsoft-com:vml	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rules) is located in §A.6.2. *end note*]

19.2.2.28 shapedefaults (New Shape Defaults)

This element specifies the defaults that are used when creating new shapes. These defaults are stored once per document.

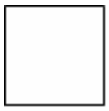
[*Example*: Consider a case in which an application chooses to store the highest shape ID it has used in the document thus far. This could be used to support the generation of new shape IDs:

```
<o:shapedefaults v:ext="edit" spidmax="1029"/>
```

end example]

Attributes	Description
allowincell (Allow in Table Cell)	Specifies whether the shape is allowed to be placed in a table cell. Default is `false`. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
ext (VML Extension	Specifies an optional value that indicates how applications that implement VML should

Attributes	Description
Handling Behavior) Namespace: urn:schemas-microsoft-com:vml	interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).

fill (Shape Fill Toggle)	Specifies whether the closed path is filled. Default is `true`. This attribute is overridden by the fill on attribute.
	
	[*Example*:
	
	`<v:shape … fill="f" fillcolor="red" …>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

fillcolor (Default Fill Color)	Specifies the default shape fill color. Default is no value. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description. The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).
spidmax (Shape ID Optional Storage)	Specifies an optional value that allows applications a mechanism for storing information they need to persist related to shape IDs. Default is 0. The possible values for this attribute are defined by the W3C XML Schema integer datatype.

stroke (Shape Stroke Toggle)	Specifies whether the path defining the shape is stroked with a solid line. The stroke element (§19.1.2.21) defines other strokes. The on attribute of the stroke element overrides this attribute. Default is `true`.
	
	[*Example*:
	
	`<v:shape … fillcolor="red"`
	`  stroke="false" strokecolor="blue"…>`
	`</v:shape>`

Attributes	Description
	 *end example*] The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

strokecolor (Shape Stroke Color)	Specifies the primary color of the brush to use to stroke the path of the shape. Default is `black`. The color attribute of the stroke element (§19.1.2.21) overrides this. Colors are typically specified as either a named color, such as `red`, or six hexadecimal digits representing the red, green and blue values of the color, such as `#00FF30`. Full details are specified in the simple type description.
	
	[*Example*:
	
	`<v:shape … strokecolor="red" …>`
	`</v:shape>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

style (Shape Styling Properties)	Specifies the CSS2 styling properties of the shape. The CSS2 (Cascading Style Sheets, Level 2) specification, a Recommendation of the World Wide Web Consortium, is available here: http://www.w3.org/TR/REC-CSS2.
	
	This attribute uses a semi-colon delimited list of “name:value” pairs, the syntax defined by section 4.1.8 of the CSS2 specification for grouped declarations (without the surrounding braces). The following tables define the allowed properties and the VML treatment of each.
	
	[*Example*:
	`<v:shape … style=’position:absolute;width:100pt;height:50pt’ …`
	`>`
	`</v:shape>`
	end example]
	
	Property | **Description**
	`flip` | Specifies that the orientation of a shape is flipped. Default is no value. Allowed values are:

Attributes	Description	
		• x - Flip along the y-axis, reversing the x-coordinates. • y - Flip along the x-axis, reversing the y-coordinates. • xy - Flip along both the y- and x-axis. • yx - Flip along both the x- and y-axis.
	`height`	Specifies the height of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`left`	Specifies the position of the left of the containing block of the shape relative to the element left of it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`margin-bottom`	Specifies the position of the bottom of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	`margin-left`	Specifies the position of the left of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are:

Attributes	Description	
		• auto - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	margin-right	Specifies the position of the right of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • auto - Default position of an element in the flow of the page. • <units> - A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	margin-top	Specifies the position of the top of the containing block of the shape relative to the shape anchor. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • auto - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
	mso-position-horizontal	Specifies the horizontal positioning data for objects in WordprocessingML documents. Default is absolute. Allowed values are: • absolute • left • center • right • inside • outside
	mso-position-horizontal-	Specifies relative horizontal position data for objects in WordprocessingML documents. This modifies the mso-

Attributes	Description	
	relative	position-horizontal property. Default is text. Allowed values are: • margin • page • text • char
	mso-position-vertical	Specifies the vertical positioning data for objects in WordprocessingML documents. Default is absolute. Allowed values are: • absolute • top • center • bottom • inside • outside
	mso-position-vertical-relative	Specifies relative vertical position data for objects in WordprocessingML documents. This modifies the mso-position-vertical property. Default is text. Allowed values are: • margin • page • text • line
	mso-wrap-distance-bottom	Specifies the distance from the bottom of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	mso-wrap-distance-left	Specifies the distance from the left side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	mso-wrap-distance-right	Specifies the distance from the right side of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape to include the margin areas. This property does not change the origin.
	mso-wrap-distance-top	Specifies the distance from the top of the shape to the text that wraps around it. Default is 0 pt. This property is different from the CSS margin property, which changes the origin of the shape

Attributes	Description
	to include the margin areas. This property does not change the origin.
`mso-wrap-edited`	Specifies whether the wrap coordinates were customized by the user. If the wrap coordinates are generated by an editor, this property is `true`; otherwise they were customized by a user. Default is `false`.
`mso-wrap-style`	Specifies the wrapping mode for text in shapes in WordprocessingML documents. Default is `square`. Allowed values are: • `square` - Wraps text inside the shape in a square. • `none` - Text does not wrap.
`position`	Specifies the kind of positioning used to place an element. Default is `static`. When the element is contained inside a group, this property shall be `absolute`. Allowed values are: • `static` - The element is positioned according to the normal flow of the page. The `top` and `left` properties are ignored. If the object is anchored inline, this value is used. • `absolute` - The element is positioned relative to the parent, using the `top` and `left` properties. • `relative` - The element is positioned according to the normal flow of the page, but the `top` and `left` properties are used. The overlap of overlapping elements is governed by the `z-index` property.
`rotation`	Specifies the angle that a shape is rotated, in degrees. Default is 0. Positive angles are clockwise.
`top`	Specifies the position of the top of the containing block of the shape relative to the element above it in the flow of the page. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's height.
`visibility`	Specifies whether a shape is displayed. Only `inherit` and `hidden` are used; any other values are mapped to `inherit`. Default is

Attributes	Description
	`inherit`. Allowed values are: • `hidden` - The shape is not visible, but is still part of the flow of the objects in the browser. Mouse events are not processed. • `inherit` - The visibility state is inherited from the parent of the shape.
	`width` — Specifies the width of the containing block of the shape. Default is 0. It is specified in CSS units or, for elements in a group, in the coordinate system of the parent element. Allowed values are: • `auto` - Default position of an element in the flow of the page. • <units>- A number with an absolute units designator (cm, mm, in, pt, pc, or px) or a relative units designator (em or ex). If no units are given, pixels (px) is assumed. • <percentage>- Value expressed as a percentage of the parent object's width.
	`z-index` — Specifies the display order of overlapping shapes. Default is 0. This property shall not be used for shapes anchored inline. Allowed values are: • `auto` - Uses the order that the shapes appear in the page, bottom to top. • <order>- A number that represents the stacking precedence. Shapes with higher numbers are placed on top of those with lower numbers. Negative numbers are allowed.
	The following properties are only used by the textbox element (§19.1.2.22): **Property** — **Description** `direction` — Specifies the direction of the text in the textbox. Default is `ltr`. This property is superceded by the `mso-direction-alt` property if that is specified. Allowed values are: • `ltr` - Text is displayed left-to-right. • `rtl` - Text is displayed right-to-left. `layout-flow` — Determines the flow of the text layout in a textbox. Default is `horizontal`. Allowed values are: • `horizontal` - Text is displayed horizontally.

Attributes	Description	
		• `vertical` - Text is displayed vertically. • `vertical-ideographic` - Ideographic text is displayed vertically. • `horizontal-ideographic` - Ideographic text is displayed horizontally.
	`mso-direction-alt`	Specifies an alternate direction for text in textboxes. Overrides the `direction` property. The only allowed value is `context`.
	`mso-fit-shape-to-text`	Specifies whether the shape stretches to fit the text in the textbox. Default is `false`.
	`mso-fit-text-to-shape`	Specifies whether the text stretches to fit the textbox. Default is `false`.
	`mso-layout-flow-alt`	Specifies the alternate layout flow for text in textboxes. This property is used instead of `layout-flow` when the layout flow is from bottom to top for non-ideographic languages. Its only value is `bottom-to-top`.
	`mso-next-textbox`	Specifies the ID of the next textbox in a series. Used to keep track of a set of linked textboxes. Default is no value.
	`mso-rotate`	Specifies a specific rotation value for text in a textbox. Default is 0. Allowed values are: • 0 • 90 • 180 • -90
	`mso-text-scale`	Specifies the scaling factor for fitting text to shapes. Default is 0. This property is only used if `mso-fit-text-to-shape` is `true`.
	`v-text-anchor`	Specifies the vertical anchoring of text in a textbox. Default is `top`. The alignment of a text anchor only becomes evident if `mso-fit-text-to-shape` is `false`. This property is different from the `vertical-align` CSS property, which is used for ideographic languages. Allowed values are: • `top` • `middle` • `bottom` • `top-center` • `middle-center` • `bottom-center` • `top-baseline` • `bottom-baseline` • `top-center-baseline` • `bottom-center-baseline`

Attributes	Description
	The following properties are only used by the textpath element (§19.1.2.23):

Property	Description
font	Specifies a compound value of font settings. Default is no value. The values are the same as those of the CSS font property. The order of definitions in the string is: font-style, font-variant, font-weight, font-size, line-height, font-family.
font-family	Specifies the family of the font. Default is no value. The values are the same as those of the CSS font-family property.
font-size	Specifies the size of the font. Default is no value. The font size is defined in points. The values are the same as those of the CSS font-size property.
font-style	Specifies the amount of slant for a font. Default is normal. The values are the same as those of the CSS font-style property. Allowed values are: • normal • italic • oblique - Treated the same as italic.
font-variant	Specifies the variant style of a font. Default is normal. The values are the same as those of the CSS font-variant property. Allowed values are: • normal • small-caps
font-weight	Specifies the thickness of the letters of the font. Default is normal. The values are the same as those of the CSS font-weight property. Allowed values are:

Value	Description
normal lighter 100 200 300 400	Treated as non-bold.
bold bolder 500 600 700 800 900	Treated as bold.

Attributes	Description
	`mso-text-shadow` — Specifies whether a shadow is applied to the text on a text path. Default is `false`. `text-decoration` — Specifies the style of text decoration. Default is none. The values are the same as those of the CSS text-decoration property. Allowed values are: • `none` • `underline` • `overline` • `line-through` • `blink` `v-rotate-letters` — Specifies whether the letters of the text are rotated counterclockwise by 90 degrees. Default is `false`. `v-same-letter-heights` — Specifies whether all letters are the same height regardless of initial case. If `true`, the lowercase letters are stretched to the height of the uppercase letters. Default is `false`. `v-text-align` — Specifies the alignment of text. Default is `left`. Allowed values are: • `left` • `right` • `center` • `justify` • `letter-justify` - Distributes the extra space between the letters. • `stretch-justify` - Stretches the letters to fill in the space. `v-text-kern` — Specifies whether kerning is turned on. Default is `false`. `v-text-reverse` — Specifies whether the layout order of rows is reversed. Default is `false`. This is used for vertical text layout. `v-text-spacing-mode` — Specifies the mode for letter spacing. Default is `tightening`. This property determines whether space is removed between each letter (`tightening`) or added between each letter (`tracking`). The amount of letter spacing change is defined by the `v-text-spacing` property. Allowed values are: • `tightening` • `tracking` `v-text-spacing` — Specifies the amount of spacing for text in 100ths of single line spacing. Default is 100. The line (§19.1.2.12), polyline (§19.1.2.15) and curve (§19.1.2.3) elements ignore the following properties:

Attributes	Description
	• top • left • width • height The following properties are not inherited by an element that references a shapetype element (§19.1.2.20) via the id attribute: • flip • height • left • margin-left • margin-top • position • rotation • top • visibility • width • z-index The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeDefaults) is located in §A.6.2. *end note*]

19.2.2.29 shapelayout (Shape Layout Properties)

This element contains child elements that store information used in the editing and layout of shapes.

Attributes	Description
ext (VML Extension Handling Behavior) Namespace: urn:schemas-microsoft-com:vml	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeLayout) is located in §A.6.2. *end note*]

19.2.2.30 signatureline (Digital Signature Line)

This element specifies a signature line in a document. A signature line provides a visual representation of a signature in a document that is digitally signed. The signature line element indicates that the VML shape in which it appears acts as that visual representation. Typically, the VML shape is an image.

[*Example*:

```
<v:shape … >
  <v:imagedata … />
  <o:signatureline v:ext="edit" id="{11979195-DE54-414B-ABD6-5F63607C648B}"
  provid="{00000000-0000-0000-0000-000000000000}" o:suggestedsigner="John Doe"
  o:suggestedsigner2="Manager" o:suggestedsigneremail=johndoe@example.com
  allowcomments="t" issignatureline="t"/>
</v:shape>
```

The signature line in the document might look like this:

end example]

Attributes	Description

addlxml (Additional Signature Information)	Specifies an optional string that is used to store additional information about the digital signature. Default is no value. [*Rationale*: Some digital signature software stores, for example, server and region information with the signature. *end rationale*]
	
	[*Example*:
	
	`<o:signatureline … o:addlxml="…">`
	`</o:signatureline>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

allowcomments (User-specified Comments Flag)	Specifies whether the user can attach comments to the signature line at signing time. Default is `false`. [*Example*:

Attributes	Description

	`<o:signatureline … allowcomments="true">`
	`</o:signatureline>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

ext (VML Extension Handling Behavior) Namespace: urn:schemas-microsoft-com:vml	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).

id (Unique ID)	Specifies a unique ID for the signature line. Default is no value.
	
	[*Example*:
	
	`<o:signatureline …`
	`id="{11979195-DE54-414B-ABD6-5F63607C648B}">`
	`</o:signatureline>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Guid simple type (Part 1, §22.9.2.4).

issignatureline (Signature Line Flag)	Specifies whether the image is a signature line. Default is `true`.
	
	[*Example*:
	
	`<o:signatureline … issignatureline="true">`
	`</o:signatureline>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

provid (Signature Provider ID)	Specifies a unique ID identifying which signature provider created the signature line. Default is no value. [*Guidance* The GUID is typically the CLSID of the provider COM add-in. *end guidance*] [*Example*: `<o:signatureline …`

Attributes	Description

	`provid="{00000000-0000-0000-0000-000000000000}">`
	`</o:signatureline>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Guid simple type (Part 1, §22.9.2.4).

showsigndate (Show Signed Date Flag)	Specifies whether the signed signature line image generated should include the date of signing. Default is `true`.
	
	[*Example*:
	
	`<o:signatureline … showsigndate="false">`
	`</o:signatureline>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

signinginstructions (Instructions for Signing)	Specifies text shown to the user at signing time. Default is no value.
	
	[*Example*:
	
	`<o:signatureline … o:signinginstructions="Sign here">`
	`</o:signatureline>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

signinginstructions set (Use Signing Instructions Flag)	Specifies whether there is data set in the signinginstructions attribute. Default is `false`.
	
	[*Example*:
	
	`<o:signatureline … signinginstructionsset="true">`
	`</o:signatureline>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

sigprovurl (Signature Provider Download URL)	Specifies the URL for downloading the signature provider. Default is no value. [*Example*:

Attributes	Description

	`<o:signatureline … o:sigprovurl="http://www.example.com">`
	`</o:signatureline>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

suggestedsigner (Suggested Signer Line 1)	Specifies the first line of information of who should sign the signature line. Default is no value.
	
	[*Example*:
	
	`<o:signatureline … o:suggestedsigner="John Doe">`
	`</o:signatureline>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

suggestedsigner2 (Suggested Signer Line 2)	Specifies the second line of information of who should sign the signature line. Default is no value.
	
	[*Example*:
	
	`<o:signatureline … o:suggestedsigner2="Title">`
	`</o:signatureline>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

suggestedsigneremail (Suggested Signer E-mail Address)	Specifies the e-mail address of who should sign the signature line. Default is no value.
	
	[*Example*:
	
	`<o:signatureline …`
	`o:suggestedsigneremail="johndoe@example.com">`
	`</o:signatureline>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SignatureLine) is located in §A.6.2. *end note*]

19.2.2.31 skew (Skew Transform)

This element specifies a perspective skew effect on a shape. The skew is applied to vector graphics, not image data on the shape in picture fills or image elements. The on attribute shall be true and a permitted value assigned to the matrix attribute.

Attributes	Description
ext (VML Extension Handling Behavior) Namespace: urn:schemas-microsoft-com:vml	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).
id (Skew ID)	Specifies a name that provides a unique identifier for a skew. Default is no value. The possible values for this attribute are defined by the W3C XML Schema string datatype.
matrix (Skew Perspective Matrix)	Specifies a perspective transform of a skew. Default is "1,0,0,1,0,0". The matrix is given in the form "s_{xx}, s_{xy}, s_{yx}, s_{yy}, p_x, p_y" where s = scale and p = perspective. If the offset attribute is in absolute units then p_x, p_y are in 1/EMU units; otherwise they are an inverse fraction of the shape size. The possible values for this attribute are defined by the W3C XML Schema string datatype.
offset (Skew Offset)	Specifies the amount of x,y offset from the shape's location. Default is "2pt,2pt". Positive values are measured from the upper left of the face of the shape. Values are specified as either an absolute measurement or a fractional value of the shape's dimensions (–0.5 to +0.5). The possible values for this attribute are defined by the W3C XML Schema string datatype.
on (Skew Toggle)	Specifies whether a skew is displayed. Default is `false`. The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).
origin (Skew Origin)	Specifies the origin of the skew. Default is "0,0". Values are typically a percentage of the shape's size and range from -0.5 to +0.5. Larger

Attributes	Description
	values are allowed that give offsets as multiples of the shape's size. The possible values for this attribute are defined by the W3C XML Schema string datatype.

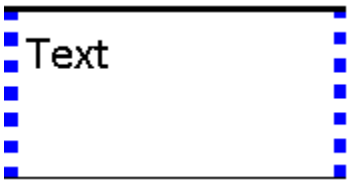
[*Note*: The W3C XML Schema definition of this element's content model (CT_Skew) is located in §A.6.2. *end note*]

19.2.2.32 top (Text Box Top Stroke)

This element specifies the stroke properties for the top border of a text box. It entirely supercedes its parent stroke element if its on attribute is true. Thus the default value of an unspecified attribute overrides a value specified in the parent. If the on attribute is `false` or not specified, the border is not shown.

[*Example*: The text box borders are set independently. Note that the bottom border does not inherit the weight from the parent stroke element.

```
<v:stroke weight="2.25pt">
  <o:left v:ext="view" dashstyle="1 1" color="blue" weight="5pt" on="t"/>
  <o:top v:ext="view" color="black" weight="2.25pt" on="t"/>
  <o:right v:ext="view" dashstyle="1 1" color="blue" weight="5pt" on="t"/>
  <o:bottom v:ext="view" color="black" on="t"/>
  <o:column v:ext="view" color="#f60" on="t"/>
</v:stroke>
```



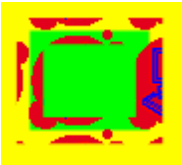
end example]

Attributes	Description

althref (Alternate Image Reference)	Specifies an alternate reference for an image in Macintosh PICT format.
	
	[*Example*:
	
	`<v:stroke … althref="myimage.pcz" … >`
	`</v:stroke>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string

Attributes	Description
	datatype.

color (Stroke Color)	Specifies the stroke color. Overrides the strokecolor attribute of a shape. Default is `black`. See the fillcolor attribute for a list of supported named colors.
	
	[*Example*: The shape stroke is blue:
	
	`<v:shape … strokecolor="red" … >`
	`  <v:stroke color="blue"/>`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

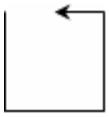
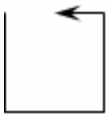
color2 (Stroke Alternate Pattern Color)	Specifies a second color for strokes, used when filltype is `pattern`. Default is no value.
	
	When a pattern fill is used for the stroke, the stroke color is used in colored parts of the source image. The color2 defines an alternate color to use in place of black in the source image.
	
	[*Example*: This unusual example is intended to demonstrate how the image and colors interact to create a patterned stroke. The yellow background shows transparency. The non-square shape and square image create an effective offset. The heavy stroke weight shows more of the image. The green shape fill shows how the stroke is overlaid on the shape.
	
	`<v:background fillcolor="yellow"/>`
	`<v:shape style="width:60;height:50"`
	`  strokecolor="red" fillcolor="lime"`
	`  path="m 0,0 l 0,1000 1000,1000 1000,0 x e">`
	`  <v:stroke filltype="pattern" weight="10pt"`
	`  src="myimage.gif" color2="blue"/>`
	`</v:shape>`
	
	  , where myimage.gif is: 
	
	end example]
	
	The possible values for this attribute are defined by the ST_ColorType simple type (§20.1.2.3).

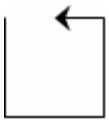
dashstyle (Stroke Dash Pattern)	Specifies the dot and dash pattern for a stroke. Default is `solid`. Pre-defined values are:

Attributes	Description

	• solid
	• shortdash
	• shortdot
	• shortdashdot
	• shortdashdotdot
	• dot
	• dash
	• longdash
	• dashdot
	• longdashdot
	• longdashdotdot
	
	A custom-defined dash pattern can also be specified using a series of numbers. These define the length of the dash (the drawn part of the stroke) and the length of the space between the dashes. The lengths are relative to the line width: a length of 1 is equal to the line width. The endcap style is applied to each dash but the arrow style is not. The string defines the length of the dash then the length of the space. This can be repeated to form complex dash styles. The string should always contain a pair of numbers; if it contains an odd number of numbers the last is disregarded. 0 implies a dot that is fourfold symmetrical (with round end caps, this is a circle).
	
	[*Example*:
	
	`<v:stroke dashstyle="0 2" weight="3pt"`
	`  endcap="round">`
	`</v:stroke>`
	
	 
	
	`<v:stroke dashstyle="longdashdotdot"`
	`  weight="2pt">`
	`</v:stroke>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

endarrow (Line End Arrowhead)	Specifies an arrowhead for the end of a line. Default is none. The path shall not be closed with the x command for the arrowhead to show. Allowed values are: • none

Attributes	Description
	• `block` • `classic` • `diamond` • `oval` • `open` [*Example*: `<v:stroke endarrow="classic"/>`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowType simple type (§19.1.3.8).
endarrowlength (Line End Arrowhead Length)	Specifies the length of the arrowhead at the end of a line. Default is `medium`. Allowed values are: • `short` • `medium` • `long` [*Example*: `<v:stroke … endarrowlength="long" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowLength simple type (§19.1.3.7).
endarrowwidth (Line End Arrowhead Width)	Specifies the width of the arrowhead at the end of a line. Default is `medium`. Allowed values are: • `narrow` • `medium` • `wide` [*Example*:

Attributes	Description
	`<v:stroke … endarrowwidth="wide" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowWidth simple type (§19.1.3.9).

endcap (Line End Cap)	Specifies the cap style for the end of a stroke. Default is `flat`. Allowed values are:
	
	• `flat`
	• `square`
	• `round`
	
	[*Example*:
	
	`<v:stroke … endcap="round" weight="10pt" … />`
	 
	`endcap="flat"`
	`endcap="square"`
	`endcap="round"`
	end example]
	
	The possible values for this attribute are defined by the ST_StrokeEndCap simple type (§19.1.3.10).

ext (VML Extension Handling Behavior) Namespace: urn:schemas-microsoft-com:vml	Specifies an optional value that indicates how applications that implement VML should interpret extensions not defined as part of the original specification of core VML. [*Rationale*: This part of the original VML specification is included to assist applications that leverage existing VML support in implementing the Office Open XML Format. *end rationale*] The possible values for this attribute are defined by the ST_Ext simple type (§19.1.3.2).
filltype (Stroke	Specifies the kind of fill used for the background of a stroke. Default is `solid`. Allowed

Attributes	Description

Image Style)	values are:
	
	• `solid` - The fill pattern is solid.
	• `tile` - The fill image is tiled.
	• `pattern` - The fill image is stretched to form a pattern.
	• `frame` - The fill image becomes a border for the shape.
	
	[*Example*:
	
	`<v:shape style="width:50;height:50"`
	`  strokecolor="red"`
	`  path="m 0,0 l 0,1000 1000,1000 1000,0 x e">`
	`  <v:stroke filltype="frame" weight="10pt"`
	`  src="border.gif"/>`
	`</v:shape>`
	
	  , where border.gif is: 
	
	end example]
	
	The possible values for this attribute are defined by the ST_FillType simple type (§19.1.3.4).

forcedash (Force Dashed Outline)	Specifies whether a dashed outline is used to draw a shape when a shape has no line or fill. Default is `false`.
	
	Used by PresentationML placeholders to draw a dashed outline when there is no line and no fill for a shape.
	
	[*Example*:
	
	`<v:shape … o:forcedash="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

href (Original Image Reference)	Specifies the URL to the original image file. Used only if the picture has been linked and embedded. Default is no value.
	
	[*Example*:
	
	`<v:fill … o:href="myimage.gif" … >`
	`</v:fill>`

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>

end example]

The possible values for this attribute are defined by the W3C XML Schema string datatype.

</td></tr>
<tr><td>imagealignshape (Stoke Image Alignment)</td><td>

Specifies the alignment of the stroke image. If `true`, the image is aligned with the shape. Otherwise, it is aligned with the containing scope. Default is `true`.

[*Example*: The top position offset shifts the image alignment relative to the containing window:

```
<v:shape fillcolor="silver"
  style="top:20;width:50;height:50"
  path="m 0,0 l 0,1000 1000,1000 1000,0 x e">
  <v:stroke imagealignshape="false" weight="20pt"
    filltype="tile" src="myimage.gif"/>
</v:shape>
```

`imagealignshape="false"`

`imagealignshape="false"`

end example]

The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

</td></tr>
<tr><td>imageaspect (Stroke Image Aspect Ratio)</td><td>

Specifies how the stroke image aspect ratio is preserved. Default is `ignore`. Allowed values are:

<table>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>ignore</td><td>Ignore aspect issues.</td></tr>
<tr><td>atleast</td><td>Image is at least as big as imagesize.</td></tr>
<tr><td>atmost</td><td>Image is no bigger than imagesize.</td></tr>
</table>

[*Example*:

```
<v:stroke filltype="frame" weight="10pt"
```

</td></tr>
</table>

Attributes	Description

	`src="border.gif" imagealignshape="true"`
	`imageaspect="atleast">`
	`</v:stroke>`
	
	  `imagealignshape="ignore"`
	  `imagealignshape="atleast"`
	  `imagealignshape="atmost"`
	
	end example]
	
	The possible values for this attribute are defined by the ST_ImageAspect simple type (§19.1.3.5).

imagesize (Stroke Image Size)	Specifies the size of the image for the stroke. Default is the size of the image. [*Example*: `<v:stroke … imagesize="10pt,10pt" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

insetpen (Inset Border From Path)	Specifies that the border shall be displayed inside of the path defining the shape, rather than along the path (the default border placement), or outside of the path as might be done with an image.
	
	[*Example*:
	
	`<v:shape … insetpen="true" … >`
	`</v:shape>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

joinstyle (Line End Join Style))	Specifies the join style for line ends. Default is `round`. • `round`

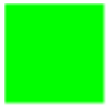
Attributes	Description

	• `bevel`
	• `miter`
	
	[*Example*:
	
	`<v:polyline strokeweight="10pt" strokecolor="navy"`
	`  points="10pt,10pt,50pt,50pt,90pt,10pt">`
	`  <v:stroke joinstyle="bevel"/>`
	`</v:polyline>`
	
	 
	`joinstyle="round"`
	
	`joinstyle="bevel"`
	
	`joinstyle="miter"`
	
	end example]
	
	The possible values for this attribute are defined by the ST_StrokeJoinStyle simple type (§19.1.3.11).

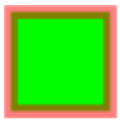
linestyle (Stroke Line Style)	Specifies the line style of the stroke. Default is `single`.
	
	• `single`
	• `thinThin`
	• `thinThick`
	• `thickThin`
	• `thickBetweenThin`
	
	[*Example*:
	
	`<v:stroke linestyle="thickThin" weight="5pt">`
	`</v:stroke>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_StrokeLineStyle simple type

Attributes	Description
	(§19.1.3.12).

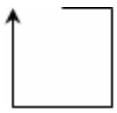
miterlimit (Miter Joint Limit)	Specifies the smoothness of the miter joint, or the maximum distance between the inner point and outer point of a joint. This number is a multiple of the thickness of the line. Default is 8.
	
	[*Example*:
	
	`<v:stroke joinstyle="miter" weight="10pt"`
	`  miterlimit="2">`
	`</v:stroke>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema decimal datatype.

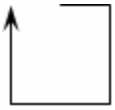
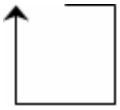
on (Stroke Toggle)	Specifies whether the stroke is displayed. Default is `true`. This attribute overrides the shape's stroke attribute.
	
	[*Example*:
	
	`<v:rect style="width:50;height:50" stroked="true"`
	`  fillcolor="lime" strokecolor="red">`
	`  <v:stroke on="false" weight="5pt"/>`
	`</v:rect>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the ST_TrueFalse simple type (§20.1.2.5).

opacity (Stroke Opacity)	Specifies the amount of transparency of a stroke. Default is 1.0.
	
	[*Example*:
	
	`<v:rect style="width:50;height:50"`
	`  fillcolor="lime" strokecolor="red">`
	`  <v:stroke weight="5pt" opacity="50%"/>`
	`</v:rect>`

Attributes	Description
	 *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

src (Stroke Image Location)	Specifies the source image to load for a stroke fill. Default is no value.
	
	[*Example*:
	
	`<v:stroke … src="myimage.gif" … >`
	`</v:stroke>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

startarrow (Line Start Arrowhead)	Specifies an arrowhead for the start of a line. Default is `none`. The path shall not be closed with the x command for the arrowhead to show. Allowed values are: • `none` • `block` • `classic` • `diamond` • `oval` • `open` [*Example*: `<v:stroke startarrow="classic"/>`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowType simple type (§19.1.3.8).
startarrowlength (Line Start Arrowhead Length)	Specifies the length of the arrowhead at the start of a line. Default is `medium`. Allowed values are: • `short` • `medium`

Attributes	Description
	• `long` [*Example*: `<v:stroke … startarrowlength="long" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowLength simple type (§19.1.3.7).
startarrowwidth (Line Start Arrowhead Width)	Specifies the width of the arrowhead at the start of a line. Default is `medium`. Allowed values are: • `narrow` • `medium` • `wide` [*Example*: `<v:stroke … startarrowwidth="wide" … />`  *end example*] The possible values for this attribute are defined by the ST_StrokeArrowWidth simple type (§19.1.3.9).

title (Stroke Title)	Specifies the title of an embedded stroke image. This is typically set to the comment property of the image, which is often blank.
	
	[*Example*:
	
	`<v:fill … o:title="alt text" … >`
	`</v:fill>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

weight (Stroke	Specifies the thickness of a stroke. Default is 1. This attribute overrides the shape's

Attributes	Description
Weight)	strokeweight attribute. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StrokeChild) is located in §A.6.2. *end note*]

19.2.3 Simple Types

The following additional simple type information in the urn:schemas-microsoft-com:office:office namespace is used for documents of a transitional conformance class.

19.2.3.1 ST_AlternateMathContentType (Alternate Math Content Type)

This simple type specifies the content type of the XML markup stored within the equationxml element.

The following values are reserved:

Value	Meaning
`officeopenxmlmath`	Specifies that the data has been stored using the Office Open XML Math syntax defined in Part 1, §22.1.
`mathml`	Specifies that the data has been stored using the MathML syntax.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AlternateMathContentType) is located in §A.6.2. *end note*]

19.2.3.2 ST_Angle (Callout Angles)

This simple type specifies values for the angle attribute of the callout element (§19.2.2.2).

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
30 (30 degrees)	30 degrees.
45 (45 degrees)	45 degrees.
60 (60 degrees)	60 degrees.
90 (90 degrees)	90 degrees.
any (Any Angle)	Unconstrained angle.
auto (Automatic Angle)	The application chooses an appropriate angle.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Angle) is located in §A.6.2. *end note*]

19.2.3.3 ST_BWMode (Black And White Modes)

This simple type specifies the ways in which a shape renders in a black and white context.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
auto (Automatic)	Use the bwpure or bwnormal attributes based on the type of output being generated.
black (Black)	Use black only.
blackTextAndLines (Black Text And Lines)	Use shades of gray, except for text and lines, which are black.
color (Color)	Do not use grayscale or black and white.
grayOutline (Gray Outlines)	Use gray and white only.
grayScale (Grayscale)	Use shades of gray only.
hide (Hide Object When Displayed in Black and White)	Do not display the object when rendering in only black and white.
highContrast (Black And White)	Use black and white only, no grays.
inverseGray (Inverse Grayscale)	Use shades of gray only, but invert light and dark grays.
lightGrayscale (Light grayscale)	Use light shades of gray only.
undrawn (Do Not Show)	Do not show the object.
white (White)	Use white only.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BWMode) is located in §A.6.2. *end note*]

19.2.3.4 ST_CalloutDrop (Callout Drop Location)

This simple type specifies location values for the drop attribute of the callout element (§19.2.2.2).

This simple type's contents are a restriction of the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CalloutDrop) is located in §A.6.2. *end note*]

19.2.3.5 ST_CalloutPlacement (Callout Placement)

This type defines location values used by the drop attribute of the callout element (§19.2.2.2).

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
bottom (Bottom placement)	Bottom of the shape.
center (Center placement)	Vertical center of the shape.
top (Top placement)	Top of the shape.
user (User-defined placement)	User-defined placement.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CalloutPlacement) is located in §A.6.2. *end note*]

19.2.3.6 ST_ColorMode (Extrusion Color Types)

This simple type specifies ways that the extrusion color is defined.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
auto (Use Shape Fill Color)	Specifies that the color of the extrusion is the same as the fill color of the shape.
custom (Use Custom Color)	Specifies that the extrusion is the color of the color attribute.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ColorMode) is located in §A.6.2. *end note*]

19.2.3.7 ST_ConnectorType (Connector Type)

This simple type specifies types of connectors.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
curved (Curved Connector)	A curved connector.
elbow (Elbow Connector)	An elbow-shaped connector.
none (No Connector)	No connector.
straight (Straight Connector)	A straight connector.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ConnectorType) is located in §A.6.2. *end note*]

19.2.3.8 ST_ConnectType (Connection Locations Type)

This simple type specifies types of connection locations.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
custom (Custom Connections)	A custom array of connection locations.
none (No)	No connection locations.
rect (Four Connections)	Standard four connection points at midpoints of top, bottom, left, and right sides.
segments (Edit Point Connections)	The edit points of the shape are used. Edit points are the black dots in a graphical editor that are used to select parts of a shape.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ConnectType) is located in §A.6.2. *end note*]

19.2.3.9 ST_ContentType (Content Type)

This simple type specifies a content type.

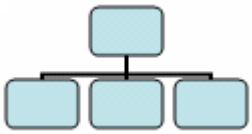
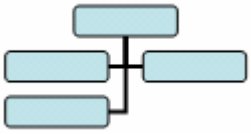
This simple type's contents are a restriction of the W3C XML Schema string datatype.

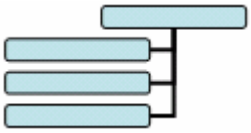
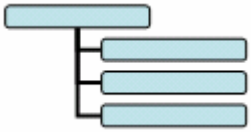
[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ContentType) is located in §A.6.2. *end note*]

19.2.3.10 ST_DiagramLayout (Diagram Layout Type)

This simple type specifies the style of automatic layout to apply to a node in a diagram.

This simple type's contents are a restriction of the W3C XML Schema integer datatype.

Enumeration Value	Description
0 (Top-down Centered)	Top-down, centered layout. 
1 (Hanging Both Sides)	Hanging, both sides layout. 
2 (Hanging Right Side)	Hanging, right side layout.

Enumeration Value	Description
	
3 (Hanging Left Side)	Hanging, left side layout. 

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DiagramLayout) is located in §A.6.2. *end note*]

19.2.3.11 ST_ExtrusionPlane (Extrusion Planes)

This simple type specifies three axis-aligned planes.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
XY (XY Plane)	The xy plane.
YZ (YZ Plane)	The yz plane.
ZX (ZX Plane)	The zx plane.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ExtrusionPlane) is located in §A.6.2. *end note*]

19.2.3.12 ST_ExtrusionRender (Extrusion Rendering Types)

This simple type specifies different rendering modes.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
boundingCube (Bounding Cube)	Rendering displays the bounding cube that contains the shape.
solid (Solid)	Rendering displays a solid shape.
wireFrame (Wireframe)	Rendering displays a wireframe shape.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ExtrusionRender) is located in §A.6.2. *end note*]

19.2.3.13 ST_ExtrusionType (Extrusion Type)

This simple type specifies types of extrusions.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
parallel (Parallel Projection)	Extrusion is rendered so that the center of projection is infinitely far away; the extrusion lines do not converge.
perspective (Perspective Projection)	Extrusion is rendered to a center of projection, which is the same as the vanishing point for unrotated objects.

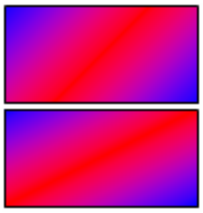
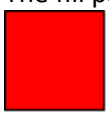
[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ExtrusionType) is located in §A.6.2. *end note*]

19.2.3.14 ST_FillType (Shape Fill Type)

This simple type specifies the types for fills applied to a shape.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
background (Use Background Fill)	Use the fill properties of the background of the object on which the shape exists, such as the page.
frame (Stretch Image to Fit)	The image is stretched to fill the shape. 
gradient (Linear Gradient)	The fill colors blend together in a linear gradient from bottom to top. 
gradientCenter (Centered Radial Gradient)	This indicates that the gradient runs across the center of the shape for a gradient that is defined as `gradientRadial` in the parent fill element (§19.1.2.5) that is defined in the VML namespace.
gradientRadial (Radial Gradient)	The fill colors blend together in a radial gradient. 
gradientUnscaled (Unscaled Gradient)	The gradient angle is not scaled relative to the aspect ratio of the shape.

Enumeration Value	Description
	[*Example*: The shapes below are twice as wide as they are tall. The first shape uses an unscaled gradient and the second uses a regular scaled gradient:  *end example*]
pattern (Image Pattern)	The image is used to create a pattern using the fill colors. 
solid (Solid Fill)	The fill pattern is a solid color. 
tile (Tiled Image)	The fill image is tiled. 

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FillType) is located in §A.6.2. *end note*]

19.2.3.15 ST_How (Alignment Type)

This simple type specifies types of alignment.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
bottom (Bottom Alignment)	Bottom vertical alignment.
center (Center Alignment)	Center horizontal alignment.
left (Left Alignment)	Left horizontal alignment.
middle (Middle Alignment)	Middle vertical alignment.
right (Right Alignment)	Right horizontal alignment.
top (Top Alignment)	Top vertical alignment.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_How) is located in §A.6.2. *end note*]

19.2.3.16 ST_HrAlign (Alignment Type)

This simple type specifies alignments for horiztonal rules.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
center (Center Alignment)	Center aligned.
left (Left Alignment)	Left aligned.
right (Right Alignment)	Right aligned.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HrAlign) is located in §A.6.2. *end note*]

19.2.3.17 ST_InsetMode (Inset Margin Type)

This simple type specifies how inner text margins are obtained.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
auto (Automatic Margins)	Inner text margins are calculated by the application.
custom (Custom Margins)	Inner text margins are specified by the shape.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_InsetMode) is located in §A.6.2. *end note*]

19.2.3.18 ST_OLEDrawAspect (Embedded Object Representations)

This simple type specifies the ways in which embedded objects are displayed in the application.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
Content (Snapshot)	The object's presentation is a picture of the contained document (provided by the embedded object server technology).
Icon (Icon)	The object's presentation is an icon.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_OLEDrawAspect) is located in §A.6.2. *end note*]

19.2.3.19 ST_OLELinkType (Embedded Object Alternate Image Request Types)

This simple type specifies the kind of image that shall be requested from the application which hosts embedded object data for a linked object. This simple type alllows any image format to be specified; however, the following values are reserved:

Enumeration Value	Description
`Bitmap`	Specifies that a bitmap should be requested.
`EnhancedMetaFile`	Specifies that a metafile (non-raster) image should be requested.
`Jpeg`	Specifies an image which should use the JPEG format.
`Picture`	Specifies that any image format can be requested. [*Example*: PNG or CGM (ISO/IEC 8632). *end example*]
`Png`	Specifies an image which should use the Portable Network Graphics format.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_OLELinkType) is located in §A.6.2. *end note*]

19.2.3.20 ST_OLEType (Embedded Connection Type)

This simple type specifies whether the embedded object is included in the package (that is, embedded) or is stored outside the package (that is, linked).

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
Embed (Embedded Object)	Embedded object.
Link (Linked Object)	Linked object.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_OLEType) is located in §A.6.2. *end note*]

19.2.3.21 ST_OLEUpdateMode (Embedded Object Update Method Type)

This simple type specifies how an embedded object is updated.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
Always (Server Application Update)	The object is updated whenever the server application using the embedded object indicates there is new data available.
OnCall (User Update)	The object is updated when the user chooses to update it.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_OLEUpdateMode) is located in §A.6.2. *end note*]

19.2.3.22 ST_RType (Rule Type)

This simple type specifies types of rules.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
align (Alignment Rule)	Alignment rule.
arc (Arc Rule)	Arc rule.
callout (Callout Rule)	Callout rule.
connector (Connector Rule)	Connector rule.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RType) is located in §A.6.2. *end note*]

19.2.3.23 ST_ScreenSize (Screen Sizes Type)

This simple type specifies screen sizes.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
1024,768 (1024x768 pixels)	1024x768 pixels.
1152,862 (1152x862 pixels)	1152x862 pixels.
544,376 (544x376 pixels)	544x376 pixels.
640,480 (640x480 pixels)	640x480 pixels.
720,512 (720x512 pixels)	720x512 pixels.
800,600 (800x600 pixels)	800x600 pixels.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ScreenSize) is located in §A.6.2. *end note*]

19.3 VML - WordprocessingML Drawing

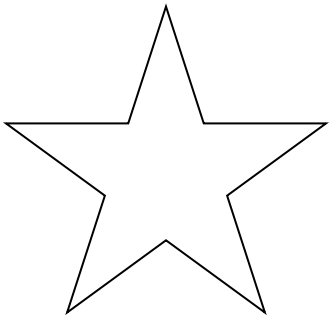
Within a WordprocessingML document, it is possible to include graphical VML objects. When these objects are present in a word processing document, it is necessary to include information about the object which is specific to their presence in a word processing document.

[*Note*: The VML format is a legacy format originally introduced with Office 2000 and is included and fully defined in ISO/IEC 29500 for backwards compatibility reasons. The DrawingML format is a newer and richer format created with the goal of eventually replacing any uses of VML in the Office Open XML formats. VML should be considered a transitional format included in Office Open XML for legacy reasons only and new applications that need a file format for drawings are strongly encouraged to use preferentially DrawingML. *end note*]

The VML WordprocessingML Drawing namespace acts in this capacity, specifying all information necessary to anchor and display VML objects within a word processing document.

All elements defined in this subclause shall only appear in a WordprocessingML document.

[*Example*: Consider a 5-point star added to a WordprocessingML document, for example:



This object allows surrounding text to wrap around its top and bottom, but not to either side, so this interaction with the surrounding document text (which is specific to a word processing document) is stored in the WordprocessingML Drawing namespace as follows:

```
<v:shape … >
  …
  <wd:wrap wd:type="topAndBottom" />
</v:shape>
```

The wrap element specifies how surrounding WordprocessingML document content must wrap around the floating VML object - in this case, by wrapping to its top and bottom extents via the type attribute value of `topAndBottom`. *end example*]

19.3.1 Table of Contents

This subclause is informative.

End of informative text.

19.3.2 Elements

The following elements comprise the contents of the urn:schemas-microsoft-com:office:word namespace:

[*Note*: As the VML format is a format provided for backward compatibility, those VML elements defined in the same urn:schemas-microsoft-com:office:word namespace remain in that namespace as it is already used by millions of documents already using VML. *end note*]

19.3.2.1 anchorlock (Anchor Location Is Locked)

This element specifies that the anchor location for this object shall not be modified at runtime when an application edits the contents of this document. [*Guidance*: An application might have automatic behaviors which reposition the anchor for a VML object based on user interaction - for example, moving it from one page to another as needed. This element must tell applications not to perform any such behaviors. *end guidance*]

If this element is omitted, then the anchor shall not be locked for the parent VML object.

[*Example*: Consider a floating VML object which must have its anchor locked at the current location. This setting is specified as follows:

```
<wd:anchorLock/>
```

The anchorLock element's presence specifies that the VML object's current anchor location must not be changed by applications editing this content. *end example*].

[*Note*: The W3C XML Schema definition of this element's content model (CT_AnchorLock) is located in §A.6.3. *end note*]

19.3.2.2 borderbottom (Bottom Border)

This element specifies the properties for the bottom border of a VML object.

Attributes	Description
shadow (Border shadow)	Specifies whether this border should be modified to create the appearance of a shadow. For the right and bottom borders, this is accomplished by duplicating the border below and right of the normal border location. For the left and top borders, this is accomplished by moving the border down and to the right of its original location. If this attribute is omitted, then the border is not given the shadow effect. [*Example*: Consider a top border which must appear with a shadow effect, resulting in the following content: `<wd:bordertop wd:shadow="true" … />` This element's shadow attribute is `true`, indicating that the shadow effect must be applied to the border. *end example*] The possible values for this attribute are defined by the ST_BorderShadow simple type (§19.3.3.1).
type (Border Style)	Specifies the style of border used on this object. See the simple type definition for a description of each border style. [*Example*: Consider a left border resulting in the following WordprocessingML: `<wd:borderleft wd:type="single" …/>` This border's type is `single`, indicating that the border style is a single line. *end example*] The possible values for this attribute are defined by the ST_BorderType simple type (§19.3.3.2).

width (Border Width)	Specifies the width of the current border.
	
	The width of this border is specified in measurements of eighths of a point, with a minimum value of two (one-fourth of a point) and a maximum value of 96 (twelve points). Any values outside this range can be reassigned to a more appropriate value.
	
	[*Example*: Consider a document with a three point wide dashed line border on all sides, resulting in the following WordprocessingML markup:
	
	`<wd:bordertop wd:type="dashed" wd:width="24" …/>`
	`<wd:borderleft wd:type="dashed" wd:width="24" …/>`
	`<wd:borderbottom wd:type="dashed" wd:width="24" …/>`
	`<wd:borderright wd:type="dashed" wd:width="24" …/>`
	
	The width attribute specifies the size in eighths of a point (24 eighths of a point = 3 points). *end example*]

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema positiveInteger datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Border) is located in §A.6.3. *end note*]

19.3.2.3 borderleft (Left Border)

This element represents the properties for the left border of a VML object.

Attributes	Description
shadow (Border shadow)	Specifies whether this border should be modified to create the appearance of a shadow. For the right and bottom borders, this is accomplished by duplicating the border below and right of the normal border location. For the left and top borders, this is accomplished by moving the border down and to the right of its original location. If this attribute is omitted, then the border is not given the shadow effect. [*Example*: Consider a top border which must appear with a shadow effect, resulting in the following content: `<wd:bordertop wd:shadow="true" … />` This element's shadow attribute is `true`, indicating that the shadow effect must be applied to the border. *end example*] The possible values for this attribute are defined by the ST_BorderShadow simple type (§19.3.3.1).
type (Border Style)	Specifies the style of border used on this object. See the simple type definition for a description of each border style. [*Example*: Consider a left border resulting in the following WordprocessingML: `<wd:borderleft wd:type="single" …/>` This border's type is `single`, indicating that the border style is a single line. *end example*] The possible values for this attribute are defined by the ST_BorderType simple type (§19.3.3.2).
width (Border Width)	Specifies the width of the current border.

Attributes	Description

	The width of this border is specified in measurements of eighths of a point, with a minimum value of two (one-fourth of a point) and a maximum value of 96 (twelve points). Any values outside this range can be reassigned to a more appropriate value.
	
	[*Example*: Consider a document with a three point wide dashed line border on all sides, resulting in the following WordprocessingML markup:
	
	`<wd:bordertop wd:type="dashed" wd:width="24" …/>`
	`<wd:borderleft wd:type="dashed" wd:width="24" …/>`
	`<wd:borderbottom wd:type="dashed" wd:width="24" …/>`
	`<wd:borderright wd:type="dashed" wd:width="24" …/>`
	
	The width attribute specifies the size in eighths of a point (24 eighths of a point = 3 points). *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema positiveInteger datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Border) is located in §A.6.3. *end note*]

19.3.2.4 borderright (Right Border)

This element specifies the properties for the right border of a VML object.

Attributes	Description
shadow (Border shadow)	Specifies whether this border should be modified to create the appearance of a shadow. For the right and bottom borders, this is accomplished by duplicating the border below and right of the normal border location. For the left and top borders, this is accomplished by moving the border down and to the right of its original location. If this attribute is omitted, then the border is not given the shadow effect. [*Example*: Consider a top border which must appear with a shadow effect, resulting in the following content: `<wd:bordertop wd:shadow="true" … />` This element's shadow attribute is `true`, indicating that the shadow effect must be applied to the border. *end example*] The possible values for this attribute are defined by the ST_BorderShadow simple type (§19.3.3.1).
type (Border Style)	Specifies the style of border used on this object.

Attributes	Description
	See the simple type definition for a description of each border style. [*Example*: Consider a left border resulting in the following WordprocessingML: `<wd:borderleft wd:type="single" …/>` This border's type is `single`, indicating that the border style is a single line. *end example*] The possible values for this attribute are defined by the ST_BorderType simple type (§19.3.3.2).

width (Border Width)	Specifies the width of the current border.
	
	The width of this border is specified in measurements of eighths of a point, with a minimum value of two (one-fourth of a point) and a maximum value of 96 (twelve points). Any values outside this range can be reassigned to a more appropriate value.
	
	[*Example*: Consider a document with a three point wide dashed line border on all sides, resulting in the following WordprocessingML markup:
	
	`<wd:bordertop wd:type="dashed" wd:width="24" …/>`
	`<wd:borderleft wd:type="dashed" wd:width="24" …/>`
	`<wd:borderbottom wd:type="dashed" wd:width="24" …/>`
	`<wd:borderright wd:type="dashed" wd:width="24" …/>`
	
	The width attribute specifies the size in eighths of a point (24 eighths of a point = 3 points). *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema positiveInteger datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Border) is located in §A.6.3. *end note*]

19.3.2.5 bordertop (Top Border)

This element specifies the properties for the top border of a VML object.

Attributes	Description
shadow (Border shadow)	Specifies whether this border should be modified to create the appearance of a shadow. For the right and bottom borders, this is accomplished by duplicating the border below and right of the normal border location. For the left and top borders, this is accomplished by moving the border down and to the right of its original location.

Attributes	Description
	If this attribute is omitted, then the border is not given the shadow effect. [*Example*: Consider a top border which must appear with a shadow effect, resulting in the following content: `<wd:bordertop wd:shadow="true" … />` This element's shadow attribute is `true`, indicating that the shadow effect must be applied to the border. *end example*] The possible values for this attribute are defined by the ST_BorderShadow simple type (§19.3.3.1).
type (Border Style)	Specifies the style of border used on this object. See the simple type definition for a description of each border style. [*Example*: Consider a left border resulting in the following WordprocessingML: `<wd:borderleft wd:type="single" …/>` This border's type is `single`, indicating that the border style is a single line. *end example*] The possible values for this attribute are defined by the ST_BorderType simple type (§19.3.3.2).

width (Border Width)	Specifies the width of the current border.
	
	The width of this border is specified in measurements of eighths of a point, with a minimum value of two (one-fourth of a point) and a maximum value of 96 (twelve points). Any values outside this range can be reassigned to a more appropriate value.
	
	[*Example*: Consider a document with a three point wide dashed line border on all sides, resulting in the following WordprocessingML markup:
	
	`<wd:bordertop wd:type="dashed" wd:width="24" …/>`
	`<wd:borderleft wd:type="dashed" wd:width="24" …/>`
	`<wd:borderbottom wd:type="dashed" wd:width="24" …/>`
	`<wd:borderright wd:type="dashed" wd:width="24" …/>`
	
	The width attribute specifies the size in eighths of a point (24 eighths of a point = 3 points). *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema positiveInteger datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Border) is located in §A.6.3. *end note*]

19.3.2.6 wrap (Text Wrapping)

This element specifies the type of text wrapping which should be allowed around the contents of this VML object.

If this element is omitted, then no text wrapping shall be performed (i.e. the object shall be presented in line with text).

[*Example*: Consider the following VML object:

```
<v:shape … >
  …
  <wd:wrap wd:type="square" />
</v:shape>
```

The wrap element specifies how surrounding WordprocessingML document content must wrap around the floating VML object - in this case, by wrapping around its extents in a square via the type attribute value of `square`. *end example*].

Attributes	Description
anchorx (Horizontal Positioning Base)	Specifies the base object from which the horizontal positioning of the object should be calculated. A VML object can be horizontally positioned relative to: • The vertical edge of the page before any runs of text (the left edge for left-to-right paragraphs, the right edge for right-to-left paragraphs) • The vertical edge of the text margin before any runs of text (the left edge for left-to-right paragraphs, the right edge for right-to-left paragraphs) • The vertical edge of the text in the paragraph containing the VML object • The position of anchor for the floating VML object in the text. If this attribute is omitted, then its value shall be assumed to be page. [*Example*: Consider a VML object which should be positioned relative to the page edges, which is specified as follows: `<wd:wrap wd:anchorx="page" wd:anchory="page" />` The anchorx attribute specifies that horizontal anchoring is relative to the edge of the page. *end example*] The possible values for this attribute are defined by the ST_HorizontalAnchor simple type (§19.3.3.3).
anchory (Vertical	Specifies the base object from which the vertical positioning of the object should be

Attributes	Description
Positioning Base)	calculated. A VML object can be vertically positioned relative to: • The horizontal top edge of the page • The horizontal edge of the top text margin before any runs of text • The horizontal top edge of line containing the VML object • The horizontal top edge of the paragraph containing the text. If this attribute is omitted, then its value shall be assumed to be page. [*Example*: Consider a VML object which should be positioned relative to the page edges, which is specified as follows: `<wd:wrap wd:anchorx="page" wd:anchory="page" />` The anchory attribute specifies that horizontal anchoring is relative to the edge of the page. *end example*] The possible values for this attribute are defined by the ST_VerticalAnchor simple type (§19.3.3.4).
side (Wrapping side)	Specifies how text shall wrap around the object's left and right sides. [*Example*: Consider a floating DrawingML object which must allow text to wrap around its left side only. This setting is specified as follows: `<wd:wrap side="left" … />` The side attribute value of `left` specifies that text must only wrap around the left side of the object. *end example*] The possible values for this attribute are defined by the ST_WrapSide simple type (§19.3.3.5).

type (Wrapping type)	Specifies the type of wrapping - see the simple type definition for a description of each type.
	
	[*Example*: Consider the following VML object:
	
	`<v:shape … >`
	`  …`
	`  <wd:wrap wd:type="topAndBottom" />`
	`</v:shape>`
	
	The wrap element specifies how surrounding WordprocessingML document content must wrap around the floating VML object - in this case, by wrapping around its top and bottom extents via the type attribute value of `topAndBottom`. *end example*]

Attributes	Description
	The possible values for this attribute are defined by the ST_WrapType simple type (§19.3.3.6).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Wrap) is located in §A.6.3. *end note*]

19.3.3 Simple Types

The following additional simple type information in the urn:schemas-microsoft-com:office:word namespace is used for documents of a transitional conformance class.

19.3.3.1 ST_BorderShadow (Border Shadow Type)

This simple type specifies logical true and false values.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
f (False)	Logical false.
false (False)	Logical false.
t (True)	Logical true.
true (True)	Logical true.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BorderShadow) is located in §A.6.3. *end note*]

19.3.3.2 ST_BorderType (Border Type)

This type defines which types of borders are supported.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
dash (pecifies a line border consisting of a dashed line around the parent object.)	Specifies a line border consisting of a dashed line around the parent object.
dashDotDot (Dash Dot Dot Border)	Specifies a line border consisting of a alternating dotted, dotted, dashed line around the parent object.
dashDotStroked (Stroked Dash Dot Border)	Specifies a line border consisting of a line with a series of alternating thin and thick strokes around the parent object.
dashedSmall (Small Dash Border)	Specifies a line border consisting of a dashed line with

Enumeration Value	Description
	small gaps around the parent object.
dot (Dotted Border)	Specifies a line border consisting of a dotted line around the parent object.
dotDash (Dot Dash Border)	Specifies a line border consisting of a alternating dotted and dashed line around the parent object.
double (Double Line Border)	Specifies a line border consisting of a double line around the parent object.
doubleWave (Double Wavy Lines Border)	Specifies a line border consisting of a double wavy line around the parent object.
hairline (Hairline Border)	Specifies a line border consisting of a very thin line.
HTMLInset (Inset Border)	Specifies a line border consisting of an inset set of lines around the parent object. [*Example*: *end example*]
HTMLOutset (Outset Border)	Specifies a line border consisting of an outset set of lines around the parent object. [*Example*: *end example*]
none (No Border)	Specifies that no border shall be applied to the current item.
single (Single Line Border)	Specifies a line border consisting of a single line around the parent object.
thick (Thick Line Border)	Specifies a line border consisting of a single line around the parent object.
thickBetweenThin (Thin-thick-thin Border)	Specifies a line border consisting of a thick line contained within a thin line with a medium sized intermediate gap around the parent object.
thickBetweenThinLarge (Large thin-thick-thin Border)	Specifies a line border consisting of a thin line contained within a thick line, contained within a thin line with a medium sized intermediate gap around the parent object.
thickBetweenThinSmall (Small thin-thick-thin Lines Border)	Specifies a line border consisting of a thin line contained within a thick line, contained within a thin

Enumeration Value	Description
	line with a small intermediate gap around the parent object.
thickThin (Thick Thin Line Border)	Specifies a line border consisting of a thick line contained within a thin line with a medium sized intermediate gap around the parent object.
thickThinLarge (Thick Thin Large Gap Border)	Specifies a line border consisting of a thick line contained within a thin line with a large sized intermediate gap around the parent object.
thickThinSmall (Small thick-thin lines border)	Specifies a line border consisting of a thick line contained within a thin line with a small intermediate gap around the parent object.
thinThick (Thin Thick Line Border)	Specifies a line border consisting of a thin line contained within a thick line contained within a thick thin with a medium sized intermediate gap between each around the parent object.
thinThickLarge (Thin Thick Large Gap Border)	Specifies a line border consisting of a thin line contained within a thick line contained within a thick thin with a large sized intermediate gap between each around the parent object.
thinThickSmall (Thin Thick Small Gap Border)	Specifies a line border consisting of a thin line contained within a thick line contained within a thick thin with a small intermediate gap between each around the parent object.
threeDEmboss (3D Embossed Border)	Specifies a line border consisting of three staged gradient lines around the parent object, getting darker towards the object. [*Example*:  *end example*]
threeDEngrave (3D Engraved Border)	Specifies a line border consisting of three staged gradient lines around the parent object, getting darker away from the object. [*Example*:  *end example*]
triple (Triple Line Border)	Specifies a line border consisting of a triple line around

Enumeration Value	Description
	the parent object.
wave (Wavy Border)	Specifies a line border consisting of a wavy line around the parent object. [*Example*: 〰〰〰〰〰〰〰〰 *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BorderType) is located in §A.6.3. *end note*]

19.3.3.3 ST_HorizontalAnchor (Horizontal Anchor Type)

This simple type specifies the horizontal position to which the parent object has been anchored in the document. This anchor position shall be used as the base location to determine the final horizontal position of the object in the document.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
char (Character)	Specifies that the parent object shall be horizontally anchored based on the position of the anchor within the text flow.
margin (Margin)	Specifies that the parent object shall be horizontally anchored to the text margins. This shall be used to specify that any horizontal positioning values shall be calculated with respect to the location of the text margin.
page (Page)	Specifies that the parent object shall be horizontally anchored to the page edge. This shall be used to specify that any horizontal positioning values shall be calculated with respect to the location of the edge of the page.
text (Text)	Specifies that the parent object shall be horizontally anchored to the text extents. This shall be used to specify that any horizontal positioning values shall be calculated with respect to the location of the edge of the text in the anchor

Enumeration Value	Description
	paragraph (including text indentations on that paragraph within the text margins).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HorizontalAnchor) is located in §A.6.3. *end note*]

19.3.3.4 ST_VerticalAnchor (Vertical Anchor Type)

This simple type specifies the vertical position to which the parent object has been anchored in the document. This anchor position shall be used as the base location to determine the final vertical position of the object in the document.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
line (Line)	Specifies that the parent object shall be vertically anchored to the line on which its anchor appears. This shall be used to specify that any vertical positioning values shall be calculated with respect to the location of the top edge of the anchor's line in the anchor paragraph.
margin (Margin)	Specifies that the parent object shall be vertically anchored to the text margins. This shall be used to specify that any vertical positioning values shall be calculated with respect to the location of the text margin.
page (Page)	Specifies that the parent object shall be vertically anchored to the page edge. This shall be used to specify that any vertical positioning values shall be calculated with respect to the location of the edge of the page.
text (Text)	Specifies that the parent object shall be vertically anchored to the text extents. This shall be used to specify that any vertical positioning values shall be calculated with respect to the location of the top edge of the text in the anchor paragraph.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_VerticalAnchor) is located in §A.6.3. *end note*]

19.3.3.5 ST_WrapSide (Text Wrapping Side)

This simple type defines which sides text can wrap around a VML object.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
both (Both sides)	Wrap text on both sides.
largest (Largest side)	Wrap text on largest side.
left (Left side)	Wrap text on left side.
right (Right side)	Wrap text on right side.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_WrapSide) is located in §A.6.3. *end note*]

19.3.3.6 ST_WrapType (Text Wrapping Type)

This simple type specifies the type of text wrapping which shall be allowed around a VML object within a document.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
none (No wrapping)	Specifies that text shall not be allowed to wrap around the remaining space on each lines around this VML object.
square (Square wrapping)	Specifies that text shall be allowed to wrap around the remaining space on each line around this text frame in the document using a rectangle touching each of the object's furthest edges.
through (Through wrapping)	Specifies that text shall be allowed to wrap around the remaining space on each line around this text frame in the document, including any holes in the object.
tight (Tight wrapping)	Specifies that text shall be allowed to tightly wrap around the remaining space on each line around this text frame in the document.
topAndBottom (Top and bottom wrapping)	Specifies that text shall not be allowed to wrap around the remaining space on each lines around the VML object. Any text content shall therefore be placed on the next

Enumeration Value	Description
	line following the object which does not intersect with the object's extents.

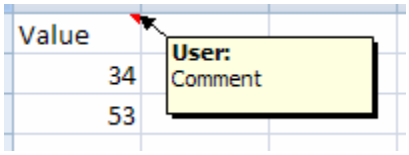
[*Note*: The W3C XML Schema definition of this simple type's content model (ST_WrapType) is located in §A.6.3. *end note*]

19.4 VML - SpreadsheetML Drawing

It is possible to attach user interface controls, such as comments, combo boxes (dropdowns) and embedded controls, to a SpreadsheetML document. VML is used to define certain aspects of the control, such as size and visual appearance. Additional information describing the control shall also be included. The VML SpreadsheetML Drawing namespace provides the additional information necessary to define the object type, settings and behavior for the control.

[*Note*: The VML format is a legacy format originally introduced with Office 2000 and is included and fully defined in ISO/IEC 29500 for backwards compatibility reasons. The DrawingML format is a newer and richer format created with the goal of eventually replacing any uses of VML in the Office Open XML formats. VML should be considered a transitional format included in Office Open XML for legacy reasons only and new applications that need a file format for drawings are strongly encouraged to use preferentially DrawingML. *end note*]

[*Example*: Assume the comment below exists on a spreadsheet:



The following defines the additional information necessary to describe the comment. The ObjectType attribute describes the object as a comment. The Anchor element defines that its edges are anchored to the first and fourth rows and the second and fourth columns. The Row and Column elements indicate that it points to the cell in the first row, first column.

```
<x:ClientData ObjectType="Note">
  <x:MoveWithCells/>
  <x:SizeWithCells/>
  <x:Anchor>1, 13, 0, 12, 2, 52, 2, 10</x:Anchor>
  <x:AutoFill>False</x:AutoFill>
  <x:Row>0</x:Row>
  <x:Column>0</x:Column>
  <x:Visible/>
</x:ClientData>
```

This additional comment data exists inside the VML shape that defines the comment object:


```
<v:shape id="_x0000_s1025" type="#_x0000_t202" style='position:absolute;margin
  left:57.75pt;margin-top:9pt;width:77.25pt;height:28.5pt;z-index:1;mso-wrap-
  style:tight' fillcolor="#ffffe1" o:insetmode="auto">
  <v:fill color2="#ffffe1"/>
  <v:shadow on="t" color="black" obscured="t"/>
  <v:path o:connecttype="none"/>
  <v:textbox style='mso-direction-alt:auto'>
    <div style='text-align:left'></div>
  </v:textbox>
  <x:ClientData ObjectType="Note"> … </x:ClientData>
</v:shape>
```

end example]

19.4.1 Table of Contents

This subclause is informative.

End of informative text.

19.4.2 Elements

The following elements comprise the contents of the urn:schemas-microsoft-com:office:excel namespace:

[*Note*: As the VML format is a format provided for backward compatibility, those VML elements defined in the same urn:schemas-microsoft-com:office:excel namespace remain in that namespace as it is already used by millions of documents already using VML. *end note*]

19.4.2.1 Accel (Primary Keyboard Accelerator)

This element specifies the primary keyboard accelerator for an object. The value is the decimal value of the Unicode character corresponding to the accelerator key. This element is used for buttons, checkboxes, radio buttons and group boxes.

[*Example*: The primary accelerator key is 'A' (65 is the decimal value for 'A' (U+0041)):

```
<x:ClientData … >
  <x:Accel>65</x:Accel>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema integer datatype.

19.4.2.2 Accel2 (Secondary Keyboard Accelerator)

This element specifies the secondary keyboard accelerator for an object. The value is the decimal value of the Unicode character corresponding to the accelerator key. This element is used for buttons, checkboxes, radio buttons and group boxes.

[*Example*: The secondary accelerator key is 'A' (65 is the decimal value for 'A' (U+0041)):

```
<x:ClientData>
  <x:Accel2>65</x:Accel2>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema integer datatype.

19.4.2.3 Anchor (Anchor)

This element specifies the anchor location for the object. This is a general-use element.

The value is a comma-separated list of data written out as: LeftColumn, LeftOffset, TopRow, TopOffset, RightColumn, RightOffset, BottomRow, BottomOffset.

Value	Description
LeftColumn	The left anchor column of the object (left-most column is 0). [*Example*:

Value	Description
	An object whose left anchor was off of the third column has a LeftColumn value of 2. *end example*]
LeftOffset	The offset of the object's left edge from the left edge of the left anchor column. This value is measured in pixels.
TopRow	The top anchor row of the object (top-most column is 0). [*Example*: An object whose top anchor was off of the fifth row has a TopRow value of 4. *end example*]
TopOffset	The offset of the object's top edge from the top edge of the top anchor row. This value is measured in pixels.
RightColumn	The right anchor column of the object (left-most column is 0). [*Example*: An object whose right anchor was off of the tenth column has a RightColumn value of 9. *end example*]
RightOffset	The offset of the object's right edge from the left edge of the right anchor column. This value is measured in pixels.
BottomRow	The bottom anchor row of the object (top-most column is 0). [*Example*: An object whose bottom anchor was off of the tenth row has a BottomRow value of 9. *end example*]
BottomOffset	The offset of the object's bottom edge from the bottom edge of the bottom anchor row. This value is measured in pixels.

[*Example*: The left side of the object is 15 pixels to the right of the left edge of the second column. The top edge is 2 pixels below the upper edge of the first row. The right side is 15 pixels to the right of the left edge of the fourth column. The bottom edge is 16 pixels below the top of the fourth row.

```
<x:ClientData>
  <x:Anchor>1, 15, 0, 2, 3, 15, 3, 16</x:Anchor>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

19.4.2.4 AutoFill (AutoFill)

This element specifies that the object's fill properties are automatically provided by the application and are not overridden with a specific fill color or style. [*Rationale*: An application can choose to display objects with certain visual properties that are appropriate to the application environment. *end rationale*] If this element is specified without a value, it is assumed to be `true`. This is a general-use element.

[*Example*:


```
<x:ClientData> …
  <x:AutoFill>False</x:AutoFill>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.5 AutoLine (AutoLine)

This element specifies that the object's line properties are automatically provided by the application and are not overridden with a specific line color, style, or width. [*Rationale*: An application can choose to display objects with certain visual properties that are appropriate to the application environment. *end rationale*] If this element is specified without a value, it is assumed to be `true`. This is a general-use element.

[*Example*:

```
<x:ClientData> …
  <x:AutoLine>False</x:AutoLine>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.6 AutoPict (Automatically Size)

This element specifies whether the object's aspect ratio is locked when rendered in different views by the application. If this element is specified without a value, it is assumed to be `true`. This is a general-use element for objects that use an image representation, denoted by the Pict value of ST_ObjectType. These objects are: embedded objects, embedded controls, cameras and signature lines.

[*Example*:

```
<x:ClientData> …
  <x:AutoPict>True</x:AutoPict>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.7 AutoScale (Font AutoScale)

This element specifies whether the object's font is automatically scaled by the application when the object is resized. If this element is specified without a value, it is assumed to be `true`. This element is used for attached text. Attached text refers to a class of objects that have text associated with them. The following values defined by the ST_ObjectType simple type are attached text objects: Button, Checkbox, Dialog, Edit, GBox, Label, Note and Radio.

[*Example*:

```
<x:ClientData> …
  <x:AutoScale>True</x:AutoScale>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

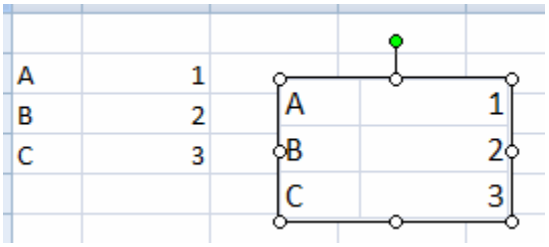
19.4.2.8 Camera (Camera Tool)

This element specifies that the object is a camera object. A camera object is a shape that is filled with a live view of a cell range in the same spreadsheet, including all applied styles. The cell range is defined by the fmlaPict element (§19.4.2.28), which shall be present. Shape properties such as the position and size of the camera object are defined by the shape. The shape shall be a rectangle. The view of the cell range is scaled vertically and horizontally to fill the rectangle exactly.

If this element is specified without a value, it is assumed to be `true`.

[*Example*:

```
<x:ClientData> …
  <x:FmlaPict>$A$2:$B$4</x:FmlaPict>
  <x:Camera>True</x:Camera>
</x:ClientData>
```



end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.9 Cancel (Cancel Button)

This element specifies that the object is a cancel button. If this element is specified without a value, it is assumed to be `true`. This element is used for buttons.

[*Example*:

```
<x:ClientData> …
  <x:Cancel/>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.10 CF (Clipboard Format)

This element specifies the clipboard format used to render the object. This is a general-use element for objects that use an image representation, such as embedded objects, embedded controls, cameras and signature lines.

[*Example*:

```
<x:ClientData> …
  <x:CF>Pict</x:CF>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_CF simple type (§19.4.3.1).

[*Note*: The W3C XML Schema definition of this element's content model (ST_CF) is located in §A.6.4. *end note*]

19.4.2.11 Checked (Checked)

This element specifies that the checkbox is checked or the radio button is selected. This element is used for checkboxes and radio buttons. Permitted values are:

Value	Description
0	Unchecked / unselected

Value	Description
1	Checked / selected
2	Mixed selection

[*Example*:

```
<x:ClientData> …
  <x:Checked>2</x:Checked>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema integer datatype.

19.4.2.12 ClientData (Attached Object Data)

This element specifies data associated with objects attached to a spreadsheet. While this element might contain any of the child elements below, only certain combinations are meaningful. The ObjectType attribute determines the kind of object the element represents and which subset of child elements is appropriate. Relevant groups are identified for each child element.

[*Example*: The following defines additional information for a comment. Its edges are anchored to the first and fourth rows and the second and fourth columns. It points to the cell in the first row, first column.

```
<x:ClientData ObjectType="Note">
  <x:MoveWithCells/>
  <x:SizeWithCells/>
  <x:Anchor>1, 15, 0, 2, 3, 15, 3, 16</x:Anchor>
  <x:AutoFill>False</x:AutoFill>
  <x:Row>0</x:Row>
  <x:Column>0</x:Column>
  <x:Visible/>
</x:ClientData>
```

end example]

[*Example*: The following defines additional information for a radio button. It is the first in a series of radio buttons and selected by default. The accelerator key is 'A' (65 is the decimal value for 'A' (U+0041)) and it is linked to the cell at column A, row 1 of the first sheet.


```
<x:ClientData ObjectType=3D"Radio">
  <x:SizeWithCells/>
  <x:AutoFill>False</x:AutoFill>
  <x:AutoLine>False</x:AutoLine>
  <x:TextVAlign>Center</x:TextVAlign>
  <x:Checked>1</x:Checked>
  <x:Accel>65</x:Accel>
  <x:FmlaLink>Sheet1!$A$1</x:FmlaLink>
  <x:FirstButton/>
</x:ClientData>
```

end example]

Attributes	Description
ObjectType (Object type)	• Specifies the kind of the object. Different sets of child elements are appropriate for different types of objects. The possible values for this attribute are defined by the ST_ObjectType simple type (§19.4.3.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ClientData) is located in §A.6.4. *end note*]

19.4.2.13 ColHidden (Comment's Column is Hidden)

This element specifies that the column of the cell to which this comment points is hidden. If this element is specified without a value, it is assumed to be `true`. This element is used for comments.

[*Example*:

```
<x:ClientData> …
  <x:ColHidden>True</x:ColHidden>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.14 Colored (Dropdown Color Toggle)

This element specifies that the dropdown is colored. If this element is specified without a value, it is assumed to be `true`. This element is used for dropdowns.

[*Example*:

```
<x:ClientData> …
  <x:Colored>True</x:Colored>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.15 Column (Comment Column Target)

This element specifies the column a comment points to. The column index is 0-based. This element is used for comments.

[*Example*:

```
<x:ClientData> …
  <x:Column>0</x:Column>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema integer datatype.

19.4.2.16 DDE (Dynamic Data Exchange)

This element specifies that the object is a DDE (Dynamic Data Exchange) link. If this element is specified without a value, it is assumed to be `true`. This is a general-use element.

[*Example*:

```
<x:ClientData> …
  <x:DDE>True</x:DDE>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.17 Default (Default Button)

This element specifies that the object is a default (OK) button . If this element is specified without a value, it is assumed to be `true`. This element is used for buttons.

[*Example*:


```
<x:ClientData> …
  <x:Default>True</x:Default>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.18 DefaultSize (Default Size Toggle)

This element specifies that the object is at its default size. If this element is specified without a value, it is assumed to be `true`. This is a general-use element.

[*Example*:

```
<x:ClientData> …
  <x:DefaultSize>True</x:DefaultSize>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.19 Disabled (Macro Disable Toggle)

This element specifies that the object cannot run an attached macro. If this element is specified without a value, it is assumed to be `true`. This is a general-use element.

[*Example*:

```
<x:ClientData> …
  <x:Disabled>True</x:Disabled>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.20 Dismiss (Dismiss Button)

This element specifies that the object is a dismiss button. If this element is specified without a value, it is assumed to be `true`. This element is used for buttons.

[*Example*:

```
<x:ClientData> …
  <x:Dismiss>True</x:Dismiss>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.21 DropLines (Dropdown Maximum Lines)

This element specifies the maximum number of lines in the dropdown before scrollbars are added. This element is used for dropdowns.

If this element is omitted, one line is shown.

[*Example*:

```
<x:ClientData> …
  <x:DropLines>8</x:DropLines>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema integer datatype.

19.4.2.22 DropStyle (Dropdown Style)

This element specifies the style of the dropdown. Allowed values are:

Value	Description
`Combo`	Standard combo box
`ComboEdit`	Editable combo box
`Simple`	Standard combo box with only the dropdown button visible when the box is not expanded

This element is used for dropdowns.

[*Example*:


```
<x:ClientData> …
  <x:DropStyle>Combo</x:DropStyle>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

19.4.2.23 Dx (Scroll Bar Width)

This element specifies the width of the scroll bar in screen pixels. This element is used for scroll bars and spinners. [*Note*: It is possible for other controls, such as combo boxes and list boxes, to use scroll bars and this element is permitted for those controls. *end note*]

[*Example*:

```
<x:ClientData> …
  <x:Dx>16</x:Dx>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema integer datatype.

19.4.2.24 FirstButton (First Radio Button)

This element specifies that the object is the first radio button in a set of radio buttons. If this element is specified without a value, it is assumed to be `true`. This element is used for radio buttons.

[*Example*:

```
<x:ClientData> …
  <x:FirstButton>True</x:FirstButton>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.25 FmlaGroup (Linked Formula - Group Box)

This element specifies the cell the object is linked to, using standard cell reference syntax. This element is used for group boxes. This overrides the FmlaLink for any radio buttons enclosed in the group box. The value in the linked cell and the index of the selected radio button are linked together. The formula syntax is described in Part 1, §18.17 of the SpreadsheetML reference.

[*Example*:

```
<x:ClientData> …
  <x:FmlaGroup>$A$1</x:FmlaGroup>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

19.4.2.26 FmlaLink (Linked Formula)

This element specifies the cell the object is linked to, using standard cell reference syntax. This element is used for checkboxes, radio buttons, scroll bars, spinners, dropdowns and list boxes. The value in the linked cell and the index of the selected item in the object are linked together. This link is ignored if the control allows multiple selections. The formula syntax is described in Part 1, §18.17 of the SpreadsheetML reference.

[*Example*:

```
<x:ClientData> …
  <x:FmlaLink>$A$4</x:FmlaLink>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

19.4.2.27 FmlaMacro (Reference to Custom Function)

This element specifies the custom function associated with the object. [*Example*: A macro script, add-in function, and so on. *end example*]

This element applies to objects defined by all values of the ST_ObjectType simple type, except: LineA, Note, RectA.

The format of this string shall be application-defined, and should be ignored if not understood.

[*Example*:

```
<x:ClientData> …
  <x:FmlaMacro>Button1_Click()</x:FmlaMacro>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

19.4.2.28 FmlaPict (Camera Source Range)

This element specifies the range of source data cells visible in the camera object (§19.4.2.8). This element is used for cameras. The formula syntax is described in Part 1, §18.17 of the SpreadsheetML reference.

This element is ignored if the Camera element is absent.

The possible values for this element are defined by the W3C XML Schema string datatype.

19.4.2.29 FmlaRange (List Items Source Range)

This element specifies the range of source data cells used to populate the list box, using standard cell reference syntax. This element is used for list boxes. The formula syntax is described in Part 1, §18.17 of the SpreadsheetML reference.

[*Example*:

```
<x:ClientData> …
  <x:FmlaRange>$A$1:$A$15</x:FmlaRange>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

19.4.2.30 FmlaTxbx (Text Formula)

This element defines the formula associated with the object's text. This element is used for attached text.

[*Example*:

```
<x:ClientData> …
  <x:FmlaTxbx>$D$9</x:FmlaTxbx>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

19.4.2.31 Help (Help Button)

This element specifies that the object is a help button. If this element is specified without a value, it is assumed to be `true`. This element is used for buttons.

[*Example*:

```
<x:ClientData> …
  <x:Help>True</x:Help>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.32 Horiz (Scroll Bar Orientation)

This element specifies that the scroll bar is horizontal. If omitted, the scroll bar is vertical. If this element is specified without a value, it is assumed to be `true`. This element is used for scroll bars and spinners.

[*Example*:

```
<x:ClientData> …
  <x:Horiz>True</x:Horiz>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.33 Inc (Scroll Bar Increment)

This element specifies the number of lines to move the scroll bar on an increment click. If omitted, the increment is 0. This element is used for scroll bars and spinners.

[*Example*:

```
<x:ClientData> …
  <x:Inc>1</x:Inc>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema integer datatype.

19.4.2.34 JustLastX (Far East Alignment Toggle)

This element specifies that Far East alignment is set for the last line in the text. Typically, justified text in Far East environments leaves the last line unjustified. Specifying this element also justifies the last line. If this element is specified without a value, it is assumed to be `true`. This element is used for attached text.

[*Example*:


```
<x:ClientData> …
  <x:JustLastX>True</x:JustLastX>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.35 LCT (Callback Type)

This element specifies the kind of list box callback. The application should use the callback to determine how to handle user actions on the list box. The only allowed value is `Normal`. This element is used for list boxes.

[*Example*:

```
<x:ClientData> …
  <x:LCT>Normal</x:LCT>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

19.4.2.36 ListItem (Non-linked List Item)

This element specifies a non-linked list item that shall be persisted with the list. This element is used for list boxes. [*Rationale*: This is a place for applications to store optional information associated with the list box. For example, an item to be shown in the list box that is not linked from another set of data. *end rationale*]

[*Example*:

```
<x:ClientData> …
  <x:ListItem>TheItem</x:ListItem>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

19.4.2.37 Locked (Lock Toggle)

This element specifies that the object is locked when the sheet is protected. If omitted, the object is assumed to be locked. If this element is specified without a value, it is assumed to be `true`. This is a general-use element.

[*Example*:

```
<x:ClientData> …
  <x:Locked>False</x:Locked>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.38 LockText (Text Lock)

This element specifies that the object's text is locked. If omitted, the object's text is assumed to be locked. If this element is specified without a value, it is assumed to be `true`. This element is used for attached text.

[*Example*:

```
<x:ClientData> …
  <x:LockText>False</x:LockText>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.39 MapOCX (Embedded Control)

This element specifies that the object is an embedded control. If this element is specified without a value, it is assumed to be `true`. This element is used for all embedded controls.

[*Example*:

```
<x:ClientData>…
  <x:MapOCX>True</x:MapOCX>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.40 Max (Scroll Bar Maximum)

This element specifies the maximum scroll bar position as the index of the list item just above the item at the top of the view when the control is scrolled all the way down. The list indexes are 1-based. If omitted, the value is assumed to be that which allows the last item to be viewed when the control is scrolled all the way down. This element is used for scroll bars and spinners.

[*Example*: Item 21 is the first item visible in the list when the object is scrolled all the way down.

```
<x:ClientData> …
  <x:Max>20</x:Max>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema integer datatype.

19.4.2.41 Min (Scroll Bar Minimum)

This element specifies the minimum scroll bar position as the index of the list item just above the item at the top of the view when the control is scrolled all the way up, typically 0. The list indexes are 1-based. If omitted, the value is assumed to be 0. This element is used for scroll bars and spinners.

[*Example*: The first item in the list is visible when the object is scrolled all the way up:

```
<x:ClientData> …
  <x:Min>0</x:Min>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema integer datatype.

19.4.2.42 MoveWithCells (Move with Cells)

This element specifies that the object moves with its underlying cells. If this element is specified without a value, it is assumed to be `true`. This is a general-use element.

[*Example*:

```
<x:ClientData> …
  <x:MoveWithCells>True</x:MoveWithCells>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.43 MultiLine (Multi-line)

This element specifies that the control is multiline. If this element is specified without a value, it is assumed to be `true`. This element is used for edit controls.

[*Example*:

```
<x:ClientData> …
  <x:Multiline>True</x:Multiline>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.44 MultiSel (Multiple Selections)

This element specifies a comma-delimited list of selected items. This element overrides the Sel element (§19.4.2.57). This element is used for list boxes that allow multiple selections. See also the SelType element (§19.4.2.58).

[*Example*:

```
<x:ClientData> …
  <x:MultiSel>3, 5, 6</x:MultiSel>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

19.4.2.45 NoThreeD (Disable 3D)

This element specifies that 3D effects are disabled. If this element is specified without a value, it is assumed to be `true`. This element is used for checkboxes, radio buttons, group boxes and scroll bars.

[*Example*:

```
<x:ClientData> …
  <x:NoThreeD>True</x:NoThreeD>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.46 NoThreeD2 (Disable 3D)

This element specifies that 3D effects are disabled. If this element is specified without a value, it is assumed to be `true`. This element is used for dropdowns and list boxes.

[*Example*:

```
<x:ClientData> …
  <x:NoThreeD2>True</x:NoThreeD2>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.47 Page (Scroll Bar Page Increment)

This element specifies the number of lines to move the scroll bar on a page click. This element is used for scroll bars and spinners.

[*Example*:

```
<x:ClientData> …
  <x:Page>9</x:Page>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema integer datatype.

19.4.2.48 PrintObject (Print Toggle)

This element specifies that the object is printed when the document is printed. If omitted, it is assumed the object prints when the document is printed. If this element is specified without a value, it is assumed to be `true`. This is a general-use element.

[*Example*:

```
<x:ClientData> …
  <x:PrintObject>False</x:PrintObject>
</x:ClientData>
```


end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.49 RecalcAlways (Recalculation Toggle)

This element defines whether the object is always included in recalculation. If this element is specified without a value, it is assumed to be `true`. This is used by controls that reference cells in the spreadsheet to update themselves when the spreadsheet changes.

[*Example*:

```
<x:ClientData> …
  <x:RecalcAlways>True</x:RecalcAlways>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.50 Row (Comment Row Target)

This element specifies the row a comment points to. The row index is 0-based. This element is used for comments.

[*Example*:

```
<x:ClientData> …
  <x:Row>0</x:Row>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema integer datatype.

19.4.2.51 RowHidden (Comment's Row is Hidden)

This element specifies that the row of the cell to which this comment points is hidden. If this element is specified without a value, it is assumed to be `true`. This element is used for comments.

[*Example*:


```
<x:ClientData> …
  <x:RowHidden>True</x:RowHidden>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.52 ScriptExtended (HTML Script Attributes)

This element specifies custom extended attributes associated with the HTML script tag. The language and id are not included in the extended attributes. If the document contains no HTML script, this element should be ignored.

[*Example*: The extended script attribute is " src="file.js"":

```
<x:ClientData> …
  <x:ScriptExtended>src=&quot;file.js&quot;</x:ScriptExtended>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

19.4.2.53 ScriptLanguage (HTML Script Language)

This element specifies the language of the custom function. If the document contains no HTML script, this element should be ignored. Allowed values are:

Value	Description
1	Java
2	Visual Basic
3	ASP
4	Other

[*Example*:

```
<x:ClientData> …
  <x:ScriptLanguage>1</x:ScriptLanguage>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema nonNegativeInteger datatype.

19.4.2.54 ScriptLocation (HTML Script Location)

This element specifies the location of the custom function. If the document contains no HTML script, this element should be ignored. Allowed values are:

Value	Description
1	Head
2	Body

[*Example*:

```
<x:ClientData> …
  <x:ScriptLocation>2</x:ScriptLocation>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema nonNegativeInteger datatype.

19.4.2.55 ScriptText (HTML Script Text)

This element specifies the script text (comment) associated with a block of HTML script in the document. If the document contains no HTML script, this element should be ignored.

[*Example*: The script text reads: "<!-- Comment -->":

```
<x:ClientData> …
  <x:ScriptText>&lt;!&#45;- Comment &#45;-&gt;</x:ScriptText>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

19.4.2.56 SecretEdit (Password Edit)

This element specifies that the object represents a password edit field. If this element is specified without a value, it is assumed to be `true`. This element is used for attached text.

[*Example*:

```
<x:ClientData> …
  <x:SecretEdit>True</x:SecretEdit>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.57 Sel (Selected Entry)

This element specifies the index of the selected item. The list indexes are 1-based. If omitted or set to a value of 0, no items are selected. This element is used for list boxes.

[*Example*:

```
<x:ClientData>…
  <x:Sel>1</x:Sel>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema integer datatype.

19.4.2.58 SelType (Selection Type)

This element specifies the kind of selection for the list box. If omitted, the control is assumed to be `Single`. Allowed values are:

Value	Description
`Single`	The listbox shall only have one selected item.
`Multi`	The listbox can have multiple items selected by clicking on each item.
`Extend`	The listbox can have multiple items selected by holding a control key and clicking on each item.

This element is used for list boxes.

[*Example*:

```
<x:ClientData> …
  <x:SelType>Single</x:SelType>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

19.4.2.59 SizeWithCells (Resize with Cells)

This element specifies that the object resizes with its underlying cells. If this element is specified without a value, it is assumed to be `true`. This is a general-use element.

[*Example*:

```
<x:ClientData> …
  <x:SizeWithCells>True</x:SizeWithCells>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.60 TextHAlign (Horizontal Text Alignment)

This element specifies the horizontal text alignment for the object. Permitted values are `Left`, `Justify`, `Center`, `Right` and `Distributed`. If omitted, the alignment is assumed to be `Left`. This element is used for attached text.

[*Example*:

```
<x:ClientData> …
  <x:TextHAlign>Right</x:TextHAlign>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

19.4.2.61 TextVAlign (Vertical Text Alignment)

This element specifies the horizontal text alignment for the object. Permitted values are `Top`, `Justify`, `Center`, `Bottom` and `Distributed`. If omitted, the alignment is assumed to be `Top`. This element is used for attached text.

[*Example*:

```
<x:ClientData> …
  <x:TextVAlign>Center</x:TextVAlign>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

19.4.2.62 UIObj (UI Object Toggle)

This element defines whether the object is a UI object. If this element is specified without a value, it is assumed to be `true`. This is a general-use element.

[*Example*:

```
<x:ClientData> …
  <x:UIObj>True</x:UIObj>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.63 Val (Scroll bar position)

This element specifies the scroll bar position as the index of the list item just above the item at the top of the view, given the current scroll position. The list indexes are 1-based. If omitted, the value is assumed to be 0. This element is used for scroll bars and spinners.

[*Example*: The first list item (item 1) is just off the top of the view. The second list item is at the top of the view.

```
<x:ClientData> …
  <x:Val>1</x:Val>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema integer datatype.

19.4.2.64 ValidIds (Valid ID)

This element specifies that the ID of a linked object is correct. This is a general-use element.

[*Example*:

```
<x:ClientData> …
  <x:ValidIds>True</x:ValidIds>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.65 Visible (Comment Visibility Toggle)

This element specifies that a comment is visible. If omitted, the comment is assumed to be invisible. If this element is specified without a value, it is assumed to be `true`. This element is used for comments.

[*Example*:

```
<x:ClientData> …
  <x:Visible>True</x:Visible>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.66 VScroll (Vertical Scroll)

This element specifies that the object has a vertical scroll. If omitted, a vertical scroll is not used. If this element is specified without a value, it is assumed to be `true`. This element is used for edit controls.

[*Example*:

```
<x:ClientData> …
  <x:VScroll>True</x:VScroll>
</x:ClientData>
```

end example]

The possible values for this element are defined by the ST_TrueFalseBlank simple type (§20.1.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

19.4.2.67 VTEdit (Validation Type)

This element specifies the kind of semantic validation to use for data input to the control. If omitted, the value is assumed to be `Text`. Permitted values are:

Value	Description
0	Text
1	Integer
2	Number
3	Reference
4	Formula

This element is used for edit controls.

[*Example*:


```
<x:ClientData> …
  <x:VTEdit>True</x:VTEdit>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema integer datatype.

19.4.2.68 WidthMin (Minimum Width)

This element specifies the smallest width allowed for the dropdown window in screen pixels. This element is used for list boxes and dropdowns.

[*Example*:

```
<x:ClientData … > …
  <x:WidthMin>78</x:WidthMin>
</x:ClientData>
```

end example]

The possible values for this element are defined by the W3C XML Schema integer datatype.

19.4.3 Simple Types

The following additional simple type information in the urn:schemas-microsoft-com:office:excel namespace is used for documents of a transitional conformance class.

19.4.3.1 ST_CF (Clipboard Format Type)

This simple type specifies the allowed clipboard formats. This simple type allows any image format to be specified; however, the following values are reserved:

Value	Description
`Bitmap`	Bitmap.
`Jpeg`	An image which should use the JPEG format.
`Pict`	Any picture format. [*Example*: SVG or JPEG. *end example*]
`PictOld`	Any picture format, but preferably one that is more likely to be supported by legacy applications.
`PictPrint`	An image rendered using the default printer's settings. This is typically of higher resolution and scaled differently compared to a picture created for on-screen rendering.
`PictScreen`	An image rendered using screen settings. This is typically lower resolution than an image created for printing.

Value	Description
Png	An image which should use the Portable Network Graphics format.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CF) is located in §A.6.4. *end note*]

19.4.3.2 ST_ObjectType (Object Type)

This simple type specifies the objects that a ClientData element can represent.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
Button (Pushbutton)	A pushbutton control.
Checkbox (Checkbox)	A checkbox control.
Dialog (Dialog)	A dialog.
Drop (Dropdown Box)	A dropdown (combo box) control.
Edit (Editable Text Field)	An editable text field control.
GBox (Group Box)	A group box control.
Group (Group)	A group of objects, such as a group of checkboxes.
Label (Label)	A label control.
LineA (Auditing Line)	A formula auditing arrow.
List (List Box)	A list control.
Movie (Movie)	A movie object in Mac format.
Note (Comment)	A comment.
Pict (Image)	A placeholder image.
Radio (Radio Button)	A radio button control.
Rect (Plain Rectangle)	A rectangle shape that is not a control.
RectA (Auditing Rectangle)	A formula auditing rectangle.
Scroll (Scroll Bar)	A scroll bar.
Shape (Plain Shape)	A general shape that is not a control.
Spin (Spin Button)	A spin button (spinner) control.

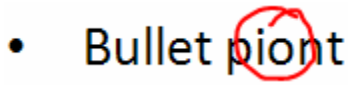
[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ObjectType) is located in §A.6.4. *end note*]

19.5 VML - PresentationML Drawing

This section describes additional information attached to VML shapes that is specific to usage with PresentationML.

[*Note*: The VML format is a legacy format originally introduced with Office 2000 and is included and fully defined in ISO/IEC 29500 for backwards compatibility reasons. The DrawingML format is a newer and richer format created with the goal of eventually replacing any uses of VML in the Office Open XML formats. VML should be considered a transitional format included in Office Open XML for legacy reasons only and new applications that need a file format for drawings are strongly encouraged to use preferentially DrawingML. *end note*]

[*Example:* Assume the following annotation was drawn on a slide during a presentation and saved into the presentation:



The red circle annotation is stored as a VML shape that is an ink annotation. For brevity, the specific path and ink data are omitted.

```
<v:shape id="_x0000_s1029" style='position:absolute;left:126pt;
  top:327.375pt;width:27.625pt;height:24.75pt' coordorigin="4445,11549"
  coordsize="973,874" path="…" filled="f" strokecolor="red"
  strokeweight="1.5pt">
  <v:stroke endcap="round"/>
  <v:path shadowok="f" o:extrusionok="f" fillok="f" insetpenok="f"/>
  <o:lock v:ext="edit" rotation="t" aspectratio="t" verticies="t" text="t"
    shapetype="t"/>
  <o:ink i="…" annotation="t"/>
  <pvml:iscomment/>
 </v:shape>
```

end example]

19.5.1 Table of Contents

This subclause is informative.

End of informative text.

19.5.2 Elements

The following elements comprise the contents of the urn:schemas-microsoft-com:office:powerpoint namespace:

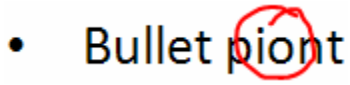
[*Note*: As the VML format is a format provided for backward compatibility, those VML elements defined in the same urn:schemas-microsoft-com:office:powerpoint namespace remain in that namespace as it is already used by millions of documents already using VML. *end note*]

19.5.2.1 iscomment (Ink Annotation Flag)

Specifies that the object was created as an ink annotation. Default is `false`. If this element is specified without a value, it is assumed to be `true`. This element is only used with PresentationML. [*Rationale* This allows an application to treat annotation ink objects as any other annotation. For example, if annotations are hidden, the application can hide the ink object. *end rationale*]

[*Example:*

```
<v:shape … >
  <o:ink … annotation="true"/>
  <pvml:iscomment/>
</v:shape>
```



end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.6.5. *end note*]

19.5.2.2 textdata (VML Diagram Text)

This element specifies optional supplementary text information associated with a legacy VML shape that is a node in a VML diagram when it cannot otherwise be stored within the DrawingML framework.

[*Note:* An application could use this to preserve a specific diagram format for backward compatibility, but it is strongly recommended to upgrade all VML shapes to DrawingML shapes. *end note*]

Attributes	Description

id (Text Reference)	Specifies the identifier that is used in conjunction with a corresponding relationship file to resolve the location of the diagram shape text.
	
	[*Example:*
	
	`<v:shape … o:dgmnodekind="0" >`
	`  <v:textbox inset="0,0,0,0"/>`
	`  <pvml:textdata id="rId1"/>`

Attributes	Description
	`</v:shape>` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rel) is located in §A.6.5. *end note*]

20. Shared MLs Reference Material

20.1 Shared Simple Types

20.1.1 Table of Contents

This subclause is informative.

End of informative text.

20.1.2 Simple Types

The following additional simple type information in the http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes namespace is used for documents of a transitional conformance class.

20.1.2.1 ST_AlgClass (Cryptographic Algorithm Classes)

This simple type specifies the possible classes of cryptographic algorithm used by protection. [*Note*: The initial version of ISO/IEC 29500 only supports a single version - `hash` - but future versions may expand this as necessary. *end note*]

[*Note*: Omitting this attribute is logically equivalent to assigning it the value `custom`. *end note*]

[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:

```
<… w:cryptAlgorithmClass="hash"
  w:cryptAlgorithmType="typeAny"
  w:cryptAlgorithmSid="1"
  w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />
```

The cryptAlgorithmClass attribute value of `hash` specifies that the algorithm used for the password is a hashing algorithm. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
custom (Custom Algorithm)	Specifies that a custom algorithm class, specified within the parent element's algIdExt attribute, generated the hash value.
hash (Hashing)	Specifies that the algorithm is a hashing function, which creates a hash value for user-supplied input that is very difficult to reverse-engineer.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AlgClass) is located in §A.7.9. *end note*]

20.1.2.2 ST_AlgType (Cryptographic Algorithm Types)

This simple type specifies the possible values for the type of cryptographic algorithm used by protection. [*Note*: The initial version of ISO/IEC 29500 only supports a single type - `typeAny` - but future versions may expand this as necessary. *end note*]

[*Note*: Omitting this attribute is logically equivalent to assigning it the value `custom`. *end note*]

[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:

```
<… w:cryptAlgorithmClass="hash"
  w:cryptAlgorithmType="typeAny"
  w:cryptAlgorithmSid="1"
  w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />
```

The cryptAlgorithmType attribute value of `typeAny` specifies that any type of algorithm may have been used for the password. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
custom (Custom Algorithm)	Specifies that a custom algorithm type, specified within the parent element's algIdExt attribute, generated the hash value.
typeAny (Any Predefined Type)	Specifies that one of the predefined cryptographic algorithms, specified by the parent element's cryptAlgorithmSid attribute, generated the hash value.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AlgType) is located in §A.7.9. *end note*]

20.1.2.3 ST_ColorType (Color Type)

This simple type specifies a color. Colors are specified in one of three ways - named color, hexadecimal RGB or color palette entry. An optional index can be stored in square brackets following the color and a space. [*Rationale:* An application might store the color's index in a system color palette using this means. *end rationale*]

A named color is specified using the name of the color. The following named colors are supported:

- `Black (#000000)`
- `Silver (#C0C0C0)`
- `Gray (#808080)`
- `White (#FFFFFF)`
- `Maroon (#800000)`
- `Red (#FF0000)`
- `Purple (#800080)`
- `Fuchsia (#FF00FF)`
- `Green (#008000)`
- `Lime (#00FF00)`
- `Olive (#808000)`
- `Yellow (#FFFF00)`
- `Navy (#000080)`
- `Blue (#0000FF)`
- `Teal (#008080)`
- `Aqua (#00FFFF)`

[*Example:*

```
<… color="red" … >
```

end example]

Hexadecimal RGB is specified using a hash symbol (#) followed by six hexadecimal characters, where each pair represents the red, green and blue component of the color.

[*Example:*

```
< … color="#5f2726" … >
```

end example]

A color palette entry is specified using the name of the color in the palette.

[*Example:*

```
<… color="buttonFace [67]" … >
```


end example]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ColorType) is located in §A.7.9. *end note*]

20.1.2.4 ST_CryptProv (Cryptographic Provider Types)

This simple type specifies the possible types of cryptographic providers which may be used.

[*Note*: Omitting this attribute is logically equivalent to assigning it the value `custom`. *end note*]

[*Example*: Consider a WordprocessingML document with the following information stored in one of its protection elements:

```
<… w:cryptProviderType="rsaAES"
  w:hash="9oN7nWkCAyEZib1RomSJTjmPpCY=" />
```

The cryptProviderType attribute value of `rsaAES` specifies that the cryptographic provider type shall be an Advanced Encryption Standard provider. *end example*

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
custom (Custom Provider)	Specifies that a custom algorithm type, specified within the parent element's algIdExt attribute, generated the hash value.
rsaAES (AES Provider)	Specifies that the provider shall support the Advanced Encryption Algorithm standard.
rsaFull (Any Provider)	Specifies that any suitable provider shall be used.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CryptProv) is located in §A.7.9. *end note*]

20.1.2.5 ST_TrueFalse (Boolean Value)

This type specifies logical true and false.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
f (False)	Logical false.
false (False)	Logical false.
t (True)	Logical true.

Enumeration Value	Description
true (True)	Logical true.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TrueFalse) is located in §A.7.9. *end note*]

20.1.2.6 ST_TrueFalseBlank (Boolean Value with Blank [False] State)

This simple type specifies a boolean value with a third state, using a blank attribute, which specifies that the value be `false`.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Enumeration Value	Description
(Blank – Logical False)	Logical false.
f (Logical False)	Logical false.
false (Logical False)	Logical false.
t (Logical True)	Logical true.
true (Logical True)	Logical true.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TrueFalseBlank) is located in §A.7.9. *end note*]

20.2 Extended Properties (Part 1, §22.2)

When used in a document of the Transitional conformance class, extended properties are stored within an Extended File Properties part with a source relationship of http://schemas.openxmlformats.org/officeDocument/2006/relationships/extended-properties.

20.3 Custom Properties (Part 1, §22.3)

When used in a document of the Transitional conformance class, custom properties are stored within a Custom File Properties part with a source relationship of http://schemas.openxmlformats.org/officeDocument/2006/relationships/custom-properties.

20.4 Changed attributes

The following attributes, which are defined in subclauses within Part 1, §22, "Shared MLs Reference Material", have different source relationships when used in documents of the Transitional conformance class:

20.4.1 Changed attribute for sources element (Part 1, §22.6.2.60)

Attributes	Description
SelectedStyle	Specifies the filename of a file which can be used to format the bibliographies and

Attributes	Description
(Selected Style)	citations within this document. If this file is of an unknown form or cannot be located, then the other attributes on this element can be used to determine the format to use. [*Example*: `<b:Sources SelectedStyle="\APA.XSL" StyleName="APA" URI="http://schemas.openxmlformats.org/bibliographicStyle/APA">` *end example*] The possible values for this attribute are defined by the ST_String simple type (Part 1, §22.9.2.13).

Annex A.
(normative)
Schemas – W3C XML Schema

This Office Open XML specification includes a family of schemas defined using the W3C XML Schema 1.0 syntax. The normative definitions of these schemas follow below, and they also reside in an accompanying file named OfficeOpenXML-XMLSchema-Transitional.zip, which is distributed in electronic form.

A.1 WordprocessingML

This schema is available in the file wml.xsd.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:m="http://schemas.openxmlformats.org/officeDocument/2006/math"
  xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
  xmlns:sl="http://schemas.openxmlformats.org/schemaLibrary/2006/main"
  xmlns:wp="http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing"
  xmlns="http://schemas.openxmlformats.org/wordprocessingml/2006/main"
  xmlns:s="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
  elementFormDefault="qualified" attributeFormDefault="qualified" blockDefault="#all"
  targetNamespace="http://schemas.openxmlformats.org/wordprocessingml/2006/main">
    <xsd:import namespace="http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing"
      schemaLocation="dml-wordprocessingDrawing.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/math"
      schemaLocation="shared-math.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
      schemaLocation="shared-relationshipReference.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/schemaLibrary/2006/main"
      schemaLocation="shared-customXmlSchemaProperties.xsd"/>
    <xsd:import namespace="http://www.w3.org/XML/1998/namespace"/>
    <xsd:complexType name="CT_Empty"/>
    <xsd:complexType name="CT_OnOff">
        <xsd:attribute name="val" type="s:ST_OnOff"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_LongHexNumber">
        <xsd:restriction base="xsd:hexBinary">
            <xsd:length value="4"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_LongHexNumber">
        <xsd:attribute name="val" type="ST_LongHexNumber" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_ShortHexNumber">
```

```
        <xsd:restriction base="xsd:hexBinary">
            <xsd:length value="2"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_UcharHexNumber">
        <xsd:restriction base="xsd:hexBinary">
            <xsd:length value="1"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Charset">
        <xsd:attribute name="val" type="ST_UcharHexNumber" use="optional"/>
        <xsd:attribute name="characterSet" type="s:ST_String" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DecimalNumberOrPercent">
        <xsd:union memberTypes="ST_UnqualifiedPercentage s:ST_Percentage"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_UnqualifiedPercentage">
        <xsd:restriction base="xsd:integer"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_DecimalNumber">
        <xsd:restriction base="xsd:integer"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_DecimalNumber">
        <xsd:attribute name="val" type="ST_DecimalNumber" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_UnsignedDecimalNumber">
        <xsd:attribute name="val" type="s:ST_UnsignedDecimalNumber" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DecimalNumberOrPrecent">
        <xsd:attribute name="val" type="ST_DecimalNumberOrPercent" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TwipsMeasure">
        <xsd:attribute name="val" type="s:ST_TwipsMeasure" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SignedTwipsMeasure">
        <xsd:union memberTypes="xsd:integer s:ST_UniversalMeasure"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_SignedTwipsMeasure">
        <xsd:attribute name="val" type="ST_SignedTwipsMeasure" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PixelsMeasure">
        <xsd:restriction base="s:ST_UnsignedDecimalNumber"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_PixelsMeasure">
        <xsd:attribute name="val" type="ST_PixelsMeasure" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_HpsMeasure">
        <xsd:union memberTypes="s:ST_UnsignedDecimalNumber s:ST_PositiveUniversalMeasure"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_HpsMeasure">
        <xsd:attribute name="val" type="ST_HpsMeasure" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SignedHpsMeasure">
```



```
        <xsd:union memberTypes="xsd:integer s:ST_UniversalMeasure"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_SignedHpsMeasure">
        <xsd:attribute name="val" type="ST_SignedHpsMeasure" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DateTime">
        <xsd:restriction base="xsd:dateTime"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_MacroName">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="33"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_MacroName">
        <xsd:attribute name="val" use="required" type="ST_MacroName"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_EighthPointMeasure">
        <xsd:restriction base="s:ST_UnsignedDecimalNumber"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PointMeasure">
        <xsd:restriction base="s:ST_UnsignedDecimalNumber"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_String">
        <xsd:attribute name="val" type="s:ST_String" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TextScale">
        <xsd:union memberTypes="ST_TextScalePercent ST_TextScaleDecimal"/>
    </xsd:simpleType>
<xsd:simpleType name="ST_TextScalePercent">
        <xsd:restriction base="xsd:string">
        <xsd:pattern value="0*(600|([0-5]?[0-9]?[0-9]))%"/>
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ST_TextScaleDecimal">
        <xsd:restriction base="xsd:integer">
        <xsd:minInclusive value="0"/>
        <xsd:maxInclusive value="600"/>
        </xsd:restriction>
</xsd:simpleType>
    <xsd:complexType name="CT_TextScale">
        <xsd:attribute name="val" type="ST_TextScale"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_HighlightColor">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="black"/>
            <xsd:enumeration value="blue"/>
            <xsd:enumeration value="cyan"/>
            <xsd:enumeration value="green"/>
            <xsd:enumeration value="magenta"/>
            <xsd:enumeration value="red"/>
            <xsd:enumeration value="yellow"/>
            <xsd:enumeration value="white"/>
            <xsd:enumeration value="darkBlue"/>
```

```
            <xsd:enumeration value="darkCyan"/>
            <xsd:enumeration value="darkGreen"/>
            <xsd:enumeration value="darkMagenta"/>
            <xsd:enumeration value="darkRed"/>
            <xsd:enumeration value="darkYellow"/>
            <xsd:enumeration value="darkGray"/>
            <xsd:enumeration value="lightGray"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Highlight">
        <xsd:attribute name="val" type="ST_HighlightColor" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_HexColorAuto">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="auto"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_HexColor">
        <xsd:union memberTypes="ST_HexColorAuto s:ST_HexColorRGB"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_Color">
        <xsd:attribute name="val" type="ST_HexColor" use="required"/>
        <xsd:attribute name="themeColor" type="ST_ThemeColor" use="optional"/>
        <xsd:attribute name="themeTint" type="ST_UcharHexNumber" use="optional"/>
        <xsd:attribute name="themeShade" type="ST_UcharHexNumber" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Lang">
        <xsd:attribute name="val" type="s:ST_Lang" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Guid">
        <xsd:attribute name="val" type="s:ST_Guid"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Underline">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="single"/>
            <xsd:enumeration value="words"/>
            <xsd:enumeration value="double"/>
            <xsd:enumeration value="thick"/>
            <xsd:enumeration value="dotted"/>
            <xsd:enumeration value="dottedHeavy"/>
            <xsd:enumeration value="dash"/>
            <xsd:enumeration value="dashedHeavy"/>
            <xsd:enumeration value="dashLong"/>
            <xsd:enumeration value="dashLongHeavy"/>
            <xsd:enumeration value="dotDash"/>
            <xsd:enumeration value="dashDotHeavy"/>
            <xsd:enumeration value="dotDotDash"/>
            <xsd:enumeration value="dashDotDotHeavy"/>
            <xsd:enumeration value="wave"/>
            <xsd:enumeration value="wavyHeavy"/>
            <xsd:enumeration value="wavyDouble"/>
            <xsd:enumeration value="none"/>
```



```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Underline">
        <xsd:attribute name="val" type="ST_Underline" use="optional"/>
        <xsd:attribute name="color" type="ST_HexColor" use="optional"/>
        <xsd:attribute name="themeColor" type="ST_ThemeColor" use="optional"/>
        <xsd:attribute name="themeTint" type="ST_UcharHexNumber" use="optional"/>
        <xsd:attribute name="themeShade" type="ST_UcharHexNumber" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TextEffect">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="blinkBackground"/>
            <xsd:enumeration value="lights"/>
            <xsd:enumeration value="antsBlack"/>
            <xsd:enumeration value="antsRed"/>
            <xsd:enumeration value="shimmer"/>
            <xsd:enumeration value="sparkle"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextEffect">
        <xsd:attribute name="val" type="ST_TextEffect" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Border">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="nil"/>
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="single"/>
            <xsd:enumeration value="thick"/>
            <xsd:enumeration value="double"/>
            <xsd:enumeration value="dotted"/>
            <xsd:enumeration value="dashed"/>
            <xsd:enumeration value="dotDash"/>
            <xsd:enumeration value="dotDotDash"/>
            <xsd:enumeration value="triple"/>
            <xsd:enumeration value="thinThickSmallGap"/>
            <xsd:enumeration value="thickThinSmallGap"/>
            <xsd:enumeration value="thinThickThinSmallGap"/>
            <xsd:enumeration value="thinThickMediumGap"/>
            <xsd:enumeration value="thickThinMediumGap"/>
            <xsd:enumeration value="thinThickThinMediumGap"/>
            <xsd:enumeration value="thinThickLargeGap"/>
            <xsd:enumeration value="thickThinLargeGap"/>
            <xsd:enumeration value="thinThickThinLargeGap"/>
            <xsd:enumeration value="wave"/>
            <xsd:enumeration value="doubleWave"/>
            <xsd:enumeration value="dashSmallGap"/>
            <xsd:enumeration value="dashDotStroked"/>
            <xsd:enumeration value="threeDEmboss"/>
            <xsd:enumeration value="threeDEngrave"/>
            <xsd:enumeration value="outset"/>
            <xsd:enumeration value="inset"/>
            <xsd:enumeration value="apples"/>
```

```
            <xsd:enumeration value="archedScallops"/>
            <xsd:enumeration value="babyPacifier"/>
            <xsd:enumeration value="babyRattle"/>
            <xsd:enumeration value="balloons3Colors"/>
            <xsd:enumeration value="balloonsHotAir"/>
            <xsd:enumeration value="basicBlackDashes"/>
            <xsd:enumeration value="basicBlackDots"/>
            <xsd:enumeration value="basicBlackSquares"/>
            <xsd:enumeration value="basicThinLines"/>
            <xsd:enumeration value="basicWhiteDashes"/>
            <xsd:enumeration value="basicWhiteDots"/>
            <xsd:enumeration value="basicWhiteSquares"/>
            <xsd:enumeration value="basicWideInline"/>
            <xsd:enumeration value="basicWideMidline"/>
            <xsd:enumeration value="basicWideOutline"/>
            <xsd:enumeration value="bats"/>
            <xsd:enumeration value="birds"/>
            <xsd:enumeration value="birdsFlight"/>
            <xsd:enumeration value="cabins"/>
            <xsd:enumeration value="cakeSlice"/>
            <xsd:enumeration value="candyCorn"/>
            <xsd:enumeration value="celticKnotwork"/>
            <xsd:enumeration value="certificateBanner"/>
            <xsd:enumeration value="chainLink"/>
            <xsd:enumeration value="champagneBottle"/>
            <xsd:enumeration value="checkedBarBlack"/>
            <xsd:enumeration value="checkedBarColor"/>
            <xsd:enumeration value="checkered"/>
            <xsd:enumeration value="christmasTree"/>
            <xsd:enumeration value="circlesLines"/>
            <xsd:enumeration value="circlesRectangles"/>
            <xsd:enumeration value="classicalWave"/>
            <xsd:enumeration value="clocks"/>
            <xsd:enumeration value="compass"/>
            <xsd:enumeration value="confetti"/>
            <xsd:enumeration value="confettiGrays"/>
            <xsd:enumeration value="confettiOutline"/>
            <xsd:enumeration value="confettiStreamers"/>
            <xsd:enumeration value="confettiWhite"/>
            <xsd:enumeration value="cornerTriangles"/>
            <xsd:enumeration value="couponCutoutDashes"/>
            <xsd:enumeration value="couponCutoutDots"/>
            <xsd:enumeration value="crazyMaze"/>
            <xsd:enumeration value="creaturesButterfly"/>
            <xsd:enumeration value="creaturesFish"/>
            <xsd:enumeration value="creaturesInsects"/>
            <xsd:enumeration value="creaturesLadyBug"/>
            <xsd:enumeration value="crossStitch"/>
            <xsd:enumeration value="cup"/>
            <xsd:enumeration value="decoArch"/>
            <xsd:enumeration value="decoArchColor"/>
            <xsd:enumeration value="decoBlocks"/>
            <xsd:enumeration value="diamondsGray"/>
```



```
            <xsd:enumeration value="doubleD"/>
            <xsd:enumeration value="doubleDiamonds"/>
            <xsd:enumeration value="earth1"/>
            <xsd:enumeration value="earth2"/>
            <xsd:enumeration value="earth3"/>
            <xsd:enumeration value="eclipsingSquares1"/>
            <xsd:enumeration value="eclipsingSquares2"/>
            <xsd:enumeration value="eggsBlack"/>
            <xsd:enumeration value="fans"/>
            <xsd:enumeration value="film"/>
            <xsd:enumeration value="firecrackers"/>
            <xsd:enumeration value="flowersBlockPrint"/>
            <xsd:enumeration value="flowersDaisies"/>
            <xsd:enumeration value="flowersModern1"/>
            <xsd:enumeration value="flowersModern2"/>
            <xsd:enumeration value="flowersPansy"/>
            <xsd:enumeration value="flowersRedRose"/>
            <xsd:enumeration value="flowersRoses"/>
            <xsd:enumeration value="flowersTeacup"/>
            <xsd:enumeration value="flowersTiny"/>
            <xsd:enumeration value="gems"/>
            <xsd:enumeration value="gingerbreadMan"/>
            <xsd:enumeration value="gradient"/>
            <xsd:enumeration value="handmade1"/>
            <xsd:enumeration value="handmade2"/>
            <xsd:enumeration value="heartBalloon"/>
            <xsd:enumeration value="heartGray"/>
            <xsd:enumeration value="hearts"/>
            <xsd:enumeration value="heebieJeebies"/>
            <xsd:enumeration value="holly"/>
            <xsd:enumeration value="houseFunky"/>
            <xsd:enumeration value="hypnotic"/>
            <xsd:enumeration value="iceCreamCones"/>
            <xsd:enumeration value="lightBulb"/>
            <xsd:enumeration value="lightning1"/>
            <xsd:enumeration value="lightning2"/>
            <xsd:enumeration value="mapPins"/>
            <xsd:enumeration value="mapleLeaf"/>
            <xsd:enumeration value="mapleMuffins"/>
            <xsd:enumeration value="marquee"/>
            <xsd:enumeration value="marqueeToothed"/>
            <xsd:enumeration value="moons"/>
            <xsd:enumeration value="mosaic"/>
            <xsd:enumeration value="musicNotes"/>
            <xsd:enumeration value="northwest"/>
            <xsd:enumeration value="ovals"/>
            <xsd:enumeration value="packages"/>
            <xsd:enumeration value="palmsBlack"/>
            <xsd:enumeration value="palmsColor"/>
            <xsd:enumeration value="paperClips"/>
            <xsd:enumeration value="papyrus"/>
            <xsd:enumeration value="partyFavor"/>
            <xsd:enumeration value="partyGlass"/>
```

```
            <xsd:enumeration value="pencils"/>
            <xsd:enumeration value="people"/>
            <xsd:enumeration value="peopleWaving"/>
            <xsd:enumeration value="peopleHats"/>
            <xsd:enumeration value="poinsettias"/>
            <xsd:enumeration value="postageStamp"/>
            <xsd:enumeration value="pumpkin1"/>
            <xsd:enumeration value="pushPinNote2"/>
            <xsd:enumeration value="pushPinNote1"/>
            <xsd:enumeration value="pyramids"/>
            <xsd:enumeration value="pyramidsAbove"/>
            <xsd:enumeration value="quadrants"/>
            <xsd:enumeration value="rings"/>
            <xsd:enumeration value="safari"/>
            <xsd:enumeration value="sawtooth"/>
            <xsd:enumeration value="sawtoothGray"/>
            <xsd:enumeration value="scaredCat"/>
            <xsd:enumeration value="seattle"/>
            <xsd:enumeration value="shadowedSquares"/>
            <xsd:enumeration value="sharksTeeth"/>
            <xsd:enumeration value="shorebirdTracks"/>
            <xsd:enumeration value="skyrocket"/>
            <xsd:enumeration value="snowflakeFancy"/>
            <xsd:enumeration value="snowflakes"/>
            <xsd:enumeration value="sombrero"/>
            <xsd:enumeration value="southwest"/>
            <xsd:enumeration value="stars"/>
            <xsd:enumeration value="starsTop"/>
            <xsd:enumeration value="stars3d"/>
            <xsd:enumeration value="starsBlack"/>
            <xsd:enumeration value="starsShadowed"/>
            <xsd:enumeration value="sun"/>
            <xsd:enumeration value="swirligig"/>
            <xsd:enumeration value="tornPaper"/>
            <xsd:enumeration value="tornPaperBlack"/>
            <xsd:enumeration value="trees"/>
            <xsd:enumeration value="triangleParty"/>
            <xsd:enumeration value="triangles"/>
            <xsd:enumeration value="triangle1"/>
            <xsd:enumeration value="triangle2"/>
            <xsd:enumeration value="triangleCircle1"/>
            <xsd:enumeration value="triangleCircle2"/>
            <xsd:enumeration value="shapes1"/>
            <xsd:enumeration value="shapes2"/>
            <xsd:enumeration value="twistedLines1"/>
            <xsd:enumeration value="twistedLines2"/>
            <xsd:enumeration value="vine"/>
            <xsd:enumeration value="waveline"/>
            <xsd:enumeration value="weavingAngles"/>
            <xsd:enumeration value="weavingBraid"/>
            <xsd:enumeration value="weavingRibbon"/>
            <xsd:enumeration value="weavingStrips"/>
            <xsd:enumeration value="whiteFlowers"/>
```



```
            <xsd:enumeration value="woodwork"/>
            <xsd:enumeration value="xIllusions"/>
            <xsd:enumeration value="zanyTriangles"/>
            <xsd:enumeration value="zigZag"/>
            <xsd:enumeration value="zigZagStitch"/>
            <xsd:enumeration value="custom"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Border">
        <xsd:attribute name="val" type="ST_Border" use="required"/>
        <xsd:attribute name="color" type="ST_HexColor" use="optional"/>
        <xsd:attribute name="themeColor" type="ST_ThemeColor" use="optional"/>
        <xsd:attribute name="themeTint" type="ST_UcharHexNumber" use="optional"/>
        <xsd:attribute name="themeShade" type="ST_UcharHexNumber" use="optional"/>
        <xsd:attribute name="sz" type="ST_EighthPointMeasure" use="optional"/>
        <xsd:attribute name="space" type="ST_PointMeasure" use="optional"/>
        <xsd:attribute name="shadow" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="frame" type="s:ST_OnOff" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Shd">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="nil"/>
            <xsd:enumeration value="clear"/>
            <xsd:enumeration value="solid"/>
            <xsd:enumeration value="horzStripe"/>
            <xsd:enumeration value="vertStripe"/>
            <xsd:enumeration value="reverseDiagStripe"/>
            <xsd:enumeration value="diagStripe"/>
            <xsd:enumeration value="horzCross"/>
            <xsd:enumeration value="diagCross"/>
            <xsd:enumeration value="thinHorzStripe"/>
            <xsd:enumeration value="thinVertStripe"/>
            <xsd:enumeration value="thinReverseDiagStripe"/>
            <xsd:enumeration value="thinDiagStripe"/>
            <xsd:enumeration value="thinHorzCross"/>
            <xsd:enumeration value="thinDiagCross"/>
            <xsd:enumeration value="pct5"/>
            <xsd:enumeration value="pct10"/>
            <xsd:enumeration value="pct12"/>
            <xsd:enumeration value="pct15"/>
            <xsd:enumeration value="pct20"/>
            <xsd:enumeration value="pct25"/>
            <xsd:enumeration value="pct30"/>
            <xsd:enumeration value="pct35"/>
            <xsd:enumeration value="pct37"/>
            <xsd:enumeration value="pct40"/>
            <xsd:enumeration value="pct45"/>
            <xsd:enumeration value="pct50"/>
            <xsd:enumeration value="pct55"/>
            <xsd:enumeration value="pct60"/>
            <xsd:enumeration value="pct62"/>
            <xsd:enumeration value="pct65"/>
            <xsd:enumeration value="pct70"/>
```

```
            <xsd:enumeration value="pct75"/>
            <xsd:enumeration value="pct80"/>
            <xsd:enumeration value="pct85"/>
            <xsd:enumeration value="pct87"/>
            <xsd:enumeration value="pct90"/>
            <xsd:enumeration value="pct95"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Shd">
        <xsd:attribute name="val" type="ST_Shd" use="required"/>
        <xsd:attribute name="color" type="ST_HexColor" use="optional"/>
        <xsd:attribute name="themeColor" type="ST_ThemeColor" use="optional"/>
        <xsd:attribute name="themeTint" type="ST_UcharHexNumber" use="optional"/>
        <xsd:attribute name="themeShade" type="ST_UcharHexNumber" use="optional"/>
        <xsd:attribute name="fill" type="ST_HexColor" use="optional"/>
        <xsd:attribute name="themeFill" type="ST_ThemeColor" use="optional"/>
        <xsd:attribute name="themeFillTint" type="ST_UcharHexNumber" use="optional"/>
        <xsd:attribute name="themeFillShade" type="ST_UcharHexNumber" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_VerticalAlignRun">
        <xsd:attribute name="val" type="s:ST_VerticalAlignRun" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FitText">
        <xsd:attribute name="val" type="s:ST_TwipsMeasure" use="required"/>
        <xsd:attribute name="id" type="ST_DecimalNumber" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Em">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="dot"/>
            <xsd:enumeration value="comma"/>
            <xsd:enumeration value="circle"/>
            <xsd:enumeration value="underDot"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Em">
        <xsd:attribute name="val" type="ST_Em" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Language">
        <xsd:attribute name="val" type="s:ST_Lang" use="optional"/>
        <xsd:attribute name="eastAsia" type="s:ST_Lang" use="optional"/>
        <xsd:attribute name="bidi" type="s:ST_Lang" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_CombineBrackets">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="round"/>
            <xsd:enumeration value="square"/>
            <xsd:enumeration value="angle"/>
            <xsd:enumeration value="curly"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_EastAsianLayout">
```



```
        <xsd:attribute name="id" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="combine" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="combineBrackets" type="ST_CombineBrackets" use="optional"/>
        <xsd:attribute name="vert" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="vertCompress" type="s:ST_OnOff" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_HeightRule">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="auto"/>
            <xsd:enumeration value="exact"/>
            <xsd:enumeration value="atLeast"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Wrap">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="auto"/>
            <xsd:enumeration value="notBeside"/>
            <xsd:enumeration value="around"/>
            <xsd:enumeration value="tight"/>
            <xsd:enumeration value="through"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_VAnchor">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="text"/>
            <xsd:enumeration value="margin"/>
            <xsd:enumeration value="page"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_HAnchor">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="text"/>
            <xsd:enumeration value="margin"/>
            <xsd:enumeration value="page"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_DropCap">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="drop"/>
            <xsd:enumeration value="margin"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_FramePr">
        <xsd:attribute name="dropCap" type="ST_DropCap" use="optional"/>
        <xsd:attribute name="lines" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="w" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="h" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="vSpace" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="hSpace" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="wrap" type="ST_Wrap" use="optional"/>
        <xsd:attribute name="hAnchor" type="ST_HAnchor" use="optional"/>
```

```
        <xsd:attribute name="vAnchor" type="ST_VAnchor" use="optional"/>
        <xsd:attribute name="x" type="ST_SignedTwipsMeasure" use="optional"/>
        <xsd:attribute name="xAlign" type="s:ST_XAlign" use="optional"/>
        <xsd:attribute name="y" type="ST_SignedTwipsMeasure" use="optional"/>
        <xsd:attribute name="yAlign" type="s:ST_YAlign" use="optional"/>
        <xsd:attribute name="hRule" type="ST_HeightRule" use="optional"/>
        <xsd:attribute name="anchorLock" type="s:ST_OnOff" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TabJc">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="clear"/>
            <xsd:enumeration value="start"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="end"/>
            <xsd:enumeration value="decimal"/>
            <xsd:enumeration value="bar"/>
            <xsd:enumeration value="num"/>
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="right"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TabTlc">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="dot"/>
            <xsd:enumeration value="hyphen"/>
            <xsd:enumeration value="underscore"/>
            <xsd:enumeration value="heavy"/>
            <xsd:enumeration value="middleDot"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TabStop">
        <xsd:attribute name="val" type="ST_TabJc" use="required"/>
        <xsd:attribute name="leader" type="ST_TabTlc" use="optional"/>
        <xsd:attribute name="pos" type="ST_SignedTwipsMeasure" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_LineSpacingRule">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="auto"/>
            <xsd:enumeration value="exact"/>
            <xsd:enumeration value="atLeast"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Spacing">
        <xsd:attribute name="before" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="beforeLines" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="beforeAutospacing" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="after" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="afterLines" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="afterAutospacing" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="line" type="ST_SignedTwipsMeasure" use="optional"/>
        <xsd:attribute name="lineRule" type="ST_LineSpacingRule" use="optional"/>
    </xsd:complexType>
```



```
    <xsd:complexType name="CT_Ind">
        <xsd:attribute name="start" type="ST_SignedTwipsMeasure" use="optional"/>
        <xsd:attribute name="startChars" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="end" type="ST_SignedTwipsMeasure" use="optional"/>
        <xsd:attribute name="endChars" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="left" type="ST_SignedTwipsMeasure" use="optional"/>
        <xsd:attribute name="leftChars" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="right" type="ST_SignedTwipsMeasure" use="optional"/>
        <xsd:attribute name="rightChars" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="hanging" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="hangingChars" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="firstLine" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="firstLineChars" type="ST_DecimalNumber" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Jc">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="start"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="end"/>
            <xsd:enumeration value="both"/>
            <xsd:enumeration value="mediumKashida"/>
            <xsd:enumeration value="distribute"/>
            <xsd:enumeration value="numTab"/>
            <xsd:enumeration value="highKashida"/>
            <xsd:enumeration value="lowKashida"/>
            <xsd:enumeration value="thaiDistribute"/>
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="right"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_JcTable">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="end"/>
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="right"/>
            <xsd:enumeration value="start"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Jc">
        <xsd:attribute name="val" type="ST_Jc" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_JcTable">
        <xsd:attribute name="val" type="ST_JcTable" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_View">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="print"/>
            <xsd:enumeration value="outline"/>
            <xsd:enumeration value="masterPages"/>
            <xsd:enumeration value="normal"/>
            <xsd:enumeration value="web"/>
```

```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_View">
        <xsd:attribute name="val" type="ST_View" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Zoom">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="fullPage"/>
            <xsd:enumeration value="bestFit"/>
            <xsd:enumeration value="textFit"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Zoom">
        <xsd:attribute name="val" type="ST_Zoom" use="optional"/>
        <xsd:attribute name="percent" type="ST_DecimalNumberOrPercent" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WritingStyle">
        <xsd:attribute name="lang" type="s:ST_Lang" use="required"/>
        <xsd:attribute name="vendorID" type="s:ST_String" use="required"/>
        <xsd:attribute name="dllVersion" type="s:ST_String" use="required"/>
        <xsd:attribute name="nlCheck" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="checkStyle" type="s:ST_OnOff" use="required"/>
        <xsd:attribute name="appName" type="s:ST_String" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Proof">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="clean"/>
            <xsd:enumeration value="dirty"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Proof">
        <xsd:attribute name="spelling" type="ST_Proof" use="optional"/>
        <xsd:attribute name="grammar" type="ST_Proof" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DocType">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_DocType">
        <xsd:attribute name="val" type="ST_DocType" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DocProtect">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="readOnly"/>
            <xsd:enumeration value="comments"/>
            <xsd:enumeration value="trackedChanges"/>
            <xsd:enumeration value="forms"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:attributeGroup name="AG_Password">
        <xsd:attribute name="algorithmName" type="s:ST_String" use="optional"/>
        <xsd:attribute name="hashValue" type="xsd:base64Binary" use="optional"/>
```



```
        <xsd:attribute name="saltValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="spinCount" type="ST_DecimalNumber" use="optional"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="AG_TransitionalPassword">
        <xsd:attribute name="cryptProviderType" type="s:ST_CryptProv"/>
        <xsd:attribute name="cryptAlgorithmClass" type="s:ST_AlgClass"/>
        <xsd:attribute name="cryptAlgorithmType" type="s:ST_AlgType"/>
        <xsd:attribute name="cryptAlgorithmSid" type="ST_DecimalNumber"/>
        <xsd:attribute name="cryptSpinCount" type="ST_DecimalNumber"/>
        <xsd:attribute name="cryptProvider" type="s:ST_String"/>
        <xsd:attribute name="algIdExt" type="ST_LongHexNumber"/>
        <xsd:attribute name="algIdExtSource" type="s:ST_String"/>
        <xsd:attribute name="cryptProviderTypeExt" type="ST_LongHexNumber"/>
        <xsd:attribute name="cryptProviderTypeExtSource" type="s:ST_String"/>
        <xsd:attribute name="hash" type="xsd:base64Binary"/>
        <xsd:attribute name="salt" type="xsd:base64Binary"/>
    </xsd:attributeGroup>
    <xsd:complexType name="CT_DocProtect">
        <xsd:attribute name="edit" type="ST_DocProtect" use="optional"/>
        <xsd:attribute name="formatting" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="enforcement" type="s:ST_OnOff"/>
        <xsd:attributeGroup ref="AG_Password"/>
        <xsd:attributeGroup ref="AG_TransitionalPassword"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_MailMergeDocType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="catalog"/>
            <xsd:enumeration value="envelopes"/>
            <xsd:enumeration value="mailingLabels"/>
            <xsd:enumeration value="formLetters"/>
            <xsd:enumeration value="email"/>
            <xsd:enumeration value="fax"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_MailMergeDocType">
        <xsd:attribute name="val" type="ST_MailMergeDocType" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_MailMergeDataType">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_MailMergeDataType">
        <xsd:attribute name="val" type="ST_MailMergeDataType" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_MailMergeDest">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="newDocument"/>
            <xsd:enumeration value="printer"/>
            <xsd:enumeration value="email"/>
            <xsd:enumeration value="fax"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_MailMergeDest">
        <xsd:attribute name="val" type="ST_MailMergeDest" use="required"/>
```

```
    </xsd:complexType>
    <xsd:simpleType name="ST_MailMergeOdsoFMDFieldType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="null"/>
            <xsd:enumeration value="dbColumn"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_MailMergeOdsoFMDFieldType">
        <xsd:attribute name="val" type="ST_MailMergeOdsoFMDFieldType" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TrackChangesView">
        <xsd:attribute name="markup" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="comments" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="insDel" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="formatting" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="inkAnnotations" type="s:ST_OnOff" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Kinsoku">
        <xsd:attribute name="lang" type="s:ST_Lang" use="required"/>
        <xsd:attribute name="val" type="s:ST_String" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TextDirection">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="tb"/>
            <xsd:enumeration value="rl"/>
            <xsd:enumeration value="lr"/>
            <xsd:enumeration value="tbV"/>
            <xsd:enumeration value="rlV"/>
            <xsd:enumeration value="lrV"/>
            <xsd:enumeration value="btLr"/>
            <xsd:enumeration value="lrTb"/>
            <xsd:enumeration value="lrTbV"/>
            <xsd:enumeration value="tbLrV"/>
            <xsd:enumeration value="tbRl"/>
            <xsd:enumeration value="tbRlV"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextDirection">
        <xsd:attribute name="val" type="ST_TextDirection" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TextAlignment">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="top"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="baseline"/>
            <xsd:enumeration value="bottom"/>
            <xsd:enumeration value="auto"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextAlignment">
        <xsd:attribute name="val" type="ST_TextAlignment" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DisplacedByCustomXml">
```



```
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="next"/>
            <xsd:enumeration value="prev"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_AnnotationVMerge">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="cont"/>
            <xsd:enumeration value="rest"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Markup">
        <xsd:attribute name="id" type="ST_DecimalNumber" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TrackChange">
        <xsd:complexContent>
            <xsd:extension base="CT_Markup">
                <xsd:attribute name="author" type="s:ST_String" use="required"/>
                <xsd:attribute name="date" type="ST_DateTime" use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_CellMergeTrackChange">
        <xsd:complexContent>
            <xsd:extension base="CT_TrackChange">
                <xsd:attribute name="vMerge" type="ST_AnnotationVMerge" use="optional"/>
                <xsd:attribute name="vMergeOrig" type="ST_AnnotationVMerge" use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_TrackChangeRange">
        <xsd:complexContent>
            <xsd:extension base="CT_TrackChange">
                <xsd:attribute name="displacedByCustomXml" type="ST_DisplacedByCustomXml"
                  use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_MarkupRange">
        <xsd:complexContent>
            <xsd:extension base="CT_Markup">
                <xsd:attribute name="displacedByCustomXml" type="ST_DisplacedByCustomXml"
                  use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_BookmarkRange">
        <xsd:complexContent>
            <xsd:extension base="CT_MarkupRange">
                <xsd:attribute name="colFirst" type="ST_DecimalNumber" use="optional"/>
                <xsd:attribute name="colLast" type="ST_DecimalNumber" use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
```

```
    </xsd:complexType>
    <xsd:complexType name="CT_Bookmark">
        <xsd:complexContent>
            <xsd:extension base="CT_BookmarkRange">
                <xsd:attribute name="name" type="s:ST_String" use="required"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_MoveBookmark">
        <xsd:complexContent>
            <xsd:extension base="CT_Bookmark">
                <xsd:attribute name="author" type="s:ST_String" use="required"/>
                <xsd:attribute name="date" type="ST_DateTime" use="required"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_Comment">
        <xsd:complexContent>
            <xsd:extension base="CT_TrackChange">
                <xsd:sequence>
                    <xsd:group ref="EG_BlockLevelElts" minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:attribute name="initials" type="s:ST_String" use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_TrackChangeNumbering">
        <xsd:complexContent>
            <xsd:extension base="CT_TrackChange">
                <xsd:attribute name="original" type="s:ST_String" use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_TblPrExChange">
        <xsd:complexContent>
            <xsd:extension base="CT_TrackChange">
                <xsd:sequence>
                    <xsd:element name="tblPrEx" type="CT_TblPrExBase" minOccurs="1"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_TcPrChange">
        <xsd:complexContent>
            <xsd:extension base="CT_TrackChange">
                <xsd:sequence>
                    <xsd:element name="tcPr" type="CT_TcPrInner" minOccurs="1"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_TrPrChange">
        <xsd:complexContent>
```



```
            <xsd:extension base="CT_TrackChange">
                <xsd:sequence>
                    <xsd:element name="trPr" type="CT_TrPrBase" minOccurs="1"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_TblGridChange">
        <xsd:complexContent>
            <xsd:extension base="CT_Markup">
                <xsd:sequence>
                    <xsd:element name="tblGrid" type="CT_TblGridBase"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_TblPrChange">
        <xsd:complexContent>
            <xsd:extension base="CT_TrackChange">
                <xsd:sequence>
                    <xsd:element name="tblPr" type="CT_TblPrBase"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_SectPrChange">
        <xsd:complexContent>
            <xsd:extension base="CT_TrackChange">
                <xsd:sequence>
                    <xsd:element name="sectPr" type="CT_SectPrBase" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_PPrChange">
        <xsd:complexContent>
            <xsd:extension base="CT_TrackChange">
                <xsd:sequence>
                    <xsd:element name="pPr" type="CT_PPrBase" minOccurs="1"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_RPrChange">
        <xsd:complexContent>
            <xsd:extension base="CT_TrackChange">
                <xsd:sequence>
                    <xsd:element name="rPr" type="CT_RPrOriginal" minOccurs="1"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_ParaRPrChange">
```

```
        <xsd:complexContent>
            <xsd:extension base="CT_TrackChange">
                <xsd:sequence>
                    <xsd:element name="rPr" type="CT_ParaRPrOriginal" minOccurs="1"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_RunTrackChange">
        <xsd:complexContent>
            <xsd:extension base="CT_TrackChange">
                <xsd:choice minOccurs="0" maxOccurs="unbounded">
                    <xsd:group ref="EG_ContentRunContent"/>
                    <xsd:group ref="m:EG_OMathMathElements"/>
                </xsd:choice>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:group name="EG_PContentMath">
      <xsd:choice>
        <xsd:group ref="EG_PContentBase" minOccurs="0" maxOccurs="unbounded" />
        <xsd:group ref="EG_ContentRunContentBase" minOccurs="0"
          maxOccurs="unbounded" />
      </xsd:choice>
    </xsd:group>
    <xsd:group name="EG_PContentBase">
      <xsd:choice>
        <xsd:element name="customXml" type="CT_CustomXmlRun"/>
        <xsd:element name="fldSimple" type="CT_SimpleField" minOccurs="0"
          maxOccurs="unbounded"/>
        <xsd:element name="hyperlink" type="CT_Hyperlink"/>
      </xsd:choice>
    </xsd:group>
    <xsd:group name="EG_ContentRunContentBase">
      <xsd:choice>
        <xsd:element name="smartTag" type="CT_SmartTagRun"/>
        <xsd:element name="sdt" type="CT_SdtRun"/>
        <xsd:group ref="EG_RunLevelElts" minOccurs="0" maxOccurs="unbounded" />
      </xsd:choice>
    </xsd:group>
    <xsd:group name="EG_CellMarkupElements">
        <xsd:choice>
            <xsd:element name="cellIns" type="CT_TrackChange" minOccurs="0"/>
            <xsd:element name="cellDel" type="CT_TrackChange" minOccurs="0"/>
            <xsd:element name="cellMerge" type="CT_CellMergeTrackChange" minOccurs="0"/>
        </xsd:choice>
    </xsd:group>
    <xsd:group name="EG_RangeMarkupElements">
        <xsd:choice>
            <xsd:element name="bookmarkStart" type="CT_Bookmark"/>
            <xsd:element name="bookmarkEnd" type="CT_MarkupRange"/>
            <xsd:element name="moveFromRangeStart" type="CT_MoveBookmark"/>
            <xsd:element name="moveFromRangeEnd" type="CT_MarkupRange"/>
```



```
            <xsd:element name="moveToRangeStart" type="CT_MoveBookmark"/>
            <xsd:element name="moveToRangeEnd" type="CT_MarkupRange"/>
            <xsd:element name="commentRangeStart" type="CT_MarkupRange"/>
            <xsd:element name="commentRangeEnd" type="CT_MarkupRange"/>
            <xsd:element name="customXmlInsRangeStart" type="CT_TrackChange"/>
            <xsd:element name="customXmlInsRangeEnd" type="CT_Markup"/>
            <xsd:element name="customXmlDelRangeStart" type="CT_TrackChange"/>
            <xsd:element name="customXmlDelRangeEnd" type="CT_Markup"/>
            <xsd:element name="customXmlMoveFromRangeStart" type="CT_TrackChange"/>
            <xsd:element name="customXmlMoveFromRangeEnd" type="CT_Markup"/>
            <xsd:element name="customXmlMoveToRangeStart" type="CT_TrackChange"/>
            <xsd:element name="customXmlMoveToRangeEnd" type="CT_Markup"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_NumPr">
        <xsd:sequence>
            <xsd:element name="ilvl" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="numId" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="numberingChange" type="CT_TrackChangeNumbering" minOccurs="0"/>
            <xsd:element name="ins" type="CT_TrackChange" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_PBdr">
        <xsd:sequence>
            <xsd:element name="top" type="CT_Border" minOccurs="0"/>
            <xsd:element name="left" type="CT_Border" minOccurs="0"/>
            <xsd:element name="bottom" type="CT_Border" minOccurs="0"/>
            <xsd:element name="right" type="CT_Border" minOccurs="0"/>
            <xsd:element name="between" type="CT_Border" minOccurs="0"/>
            <xsd:element name="bar" type="CT_Border" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Tabs">
        <xsd:sequence>
            <xsd:element name="tab" type="CT_TabStop" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_TextboxTightWrap">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="allLines"/>
            <xsd:enumeration value="firstAndLastLine"/>
            <xsd:enumeration value="firstLineOnly"/>
            <xsd:enumeration value="lastLineOnly"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextboxTightWrap">
        <xsd:attribute name="val" type="ST_TextboxTightWrap" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PPr">
        <xsd:complexContent>
            <xsd:extension base="CT_PPrBase">
                <xsd:sequence>
```

```
                    <xsd:element name="rPr" type="CT_ParaRPr" minOccurs="0"/>
                    <xsd:element name="sectPr" type="CT_SectPr" minOccurs="0"/>
                    <xsd:element name="pPrChange" type="CT_PPrChange" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_PPrBase">
        <xsd:sequence>
            <xsd:element name="pStyle" type="CT_String" minOccurs="0"/>
            <xsd:element name="keepNext" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="keepLines" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="pageBreakBefore" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="framePr" type="CT_FramePr" minOccurs="0"/>
            <xsd:element name="widowControl" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="numPr" type="CT_NumPr" minOccurs="0"/>
            <xsd:element name="suppressLineNumbers" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="pBdr" type="CT_PBdr" minOccurs="0"/>
            <xsd:element name="shd" type="CT_Shd" minOccurs="0"/>
            <xsd:element name="tabs" type="CT_Tabs" minOccurs="0"/>
            <xsd:element name="suppressAutoHyphens" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="kinsoku" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="wordWrap" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="overflowPunct" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="topLinePunct" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="autoSpaceDE" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="autoSpaceDN" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="bidi" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="adjustRightInd" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="snapToGrid" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="spacing" type="CT_Spacing" minOccurs="0"/>
            <xsd:element name="ind" type="CT_Ind" minOccurs="0"/>
            <xsd:element name="contextualSpacing" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="mirrorIndents" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="suppressOverlap" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="jc" type="CT_Jc" minOccurs="0"/>
            <xsd:element name="textDirection" type="CT_TextDirection" minOccurs="0"/>
            <xsd:element name="textAlignment" type="CT_TextAlignment" minOccurs="0"/>
            <xsd:element name="textboxTightWrap" type="CT_TextboxTightWrap" minOccurs="0"/>
            <xsd:element name="outlineLvl" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="divId" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="cnfStyle" type="CT_Cnf" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_PPrGeneral">
        <xsd:complexContent>
            <xsd:extension base="CT_PPrBase">
                <xsd:sequence>
                    <xsd:element name="pPrChange" type="CT_PPrChange" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
```



```
    <xsd:complexType name="CT_Control">
        <xsd:attribute name="name" type="s:ST_String" use="optional"/>
        <xsd:attribute name="shapeid" type="s:ST_String" use="optional"/>
        <xsd:attribute ref="r:id" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Background">
        <xsd:sequence>
            <xsd:sequence maxOccurs="unbounded">
                <xsd:any processContents="lax" namespace="urn:schemas-microsoft-com:vml" minOccurs="0"
                  maxOccurs="unbounded"/>
                <xsd:any processContents="lax" namespace="urn:schemas-microsoft-com:office:office"
                  minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:element name="drawing" type="CT_Drawing" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="color" type="ST_HexColor" use="optional"/>
        <xsd:attribute name="themeColor" type="ST_ThemeColor" use="optional"/>
        <xsd:attribute name="themeTint" type="ST_UcharHexNumber" use="optional"/>
        <xsd:attribute name="themeShade" type="ST_UcharHexNumber" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Rel">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Object">
        <xsd:sequence>
            <xsd:sequence maxOccurs="unbounded">
                <xsd:any processContents="lax" namespace="urn:schemas-microsoft-com:vml" minOccurs="0"
                  maxOccurs="unbounded"/>
                <xsd:any processContents="lax" namespace="urn:schemas-microsoft-com:office:office"
                  minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:element name="drawing" type="CT_Drawing" minOccurs="0"/>
            <xsd:choice minOccurs="0">
                <xsd:element name="control" type="CT_Control"/>
                <xsd:element name="objectLink" type="CT_ObjectLink"/>
                <xsd:element name="objectEmbed" type="CT_ObjectEmbed"/>
                <xsd:element name="movie" type="CT_Rel"/>
            </xsd:choice>
        </xsd:sequence>
        <xsd:attribute name="dxaOrig" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="dyaOrig" type="s:ST_TwipsMeasure" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Picture">
        <xsd:sequence>
            <xsd:sequence maxOccurs="unbounded">
                <xsd:any processContents="lax" namespace="urn:schemas-microsoft-com:vml" minOccurs="0"
                  maxOccurs="unbounded"/>
                <xsd:any processContents="lax" namespace="urn:schemas-microsoft-com:office:office"
                  minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:element name="movie" type="CT_Rel" minOccurs="0"/>
            <xsd:element name="control" type="CT_Control" minOccurs="0"/>
        </xsd:sequence>
```

```
    </xsd:complexType>
    <xsd:complexType name="CT_ObjectEmbed">
        <xsd:attribute name="drawAspect" type="ST_ObjectDrawAspect" use="optional"/>
        <xsd:attribute ref="r:id" use="required"/>
        <xsd:attribute name="progId" type="s:ST_String" use="optional"/>
        <xsd:attribute name="shapeId" type="s:ST_String" use="optional"/>
        <xsd:attribute name="fieldCodes" type="s:ST_String" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_ObjectDrawAspect">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="content"/>
            <xsd:enumeration value="icon"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_ObjectLink">
        <xsd:complexContent>
            <xsd:extension base="CT_ObjectEmbed">
                <xsd:attribute name="updateMode" type="ST_ObjectUpdateMode" use="required"/>
                <xsd:attribute name="lockedField" type="s:ST_OnOff" use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:simpleType name="ST_ObjectUpdateMode">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="always"/>
            <xsd:enumeration value="onCall"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Drawing">
        <xsd:choice minOccurs="1" maxOccurs="unbounded">
            <xsd:element ref="wp:anchor" minOccurs="0"/>
            <xsd:element ref="wp:inline" minOccurs="0"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_SimpleField">
        <xsd:sequence>
            <xsd:element name="fldData" type="CT_Text" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_PContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="instr" type="s:ST_String" use="required"/>
        <xsd:attribute name="fldLock" type="s:ST_OnOff"/>
        <xsd:attribute name="dirty" type="s:ST_OnOff"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_FldCharType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="begin"/>
            <xsd:enumeration value="separate"/>
            <xsd:enumeration value="end"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_InfoTextType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="text"/>
```



```
            <xsd:enumeration value="autoText"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FFHelpTextVal">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="256"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FFStatusTextVal">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="140"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FFName">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="65"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FFTextType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="regular"/>
            <xsd:enumeration value="number"/>
            <xsd:enumeration value="date"/>
            <xsd:enumeration value="currentTime"/>
            <xsd:enumeration value="currentDate"/>
            <xsd:enumeration value="calculated"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_FFTextType">
        <xsd:attribute name="val" type="ST_FFTextType" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FFName">
        <xsd:attribute name="val" type="ST_FFName"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FldChar">
        <xsd:choice>
            <xsd:element name="fldData" type="CT_Text" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ffData" type="CT_FFData" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="numberingChange" type="CT_TrackChangeNumbering" minOccurs="0"/>
        </xsd:choice>
        <xsd:attribute name="fldCharType" type="ST_FldCharType" use="required"/>
        <xsd:attribute name="fldLock" type="s:ST_OnOff"/>
        <xsd:attribute name="dirty" type="s:ST_OnOff"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Hyperlink">
        <xsd:group ref="EG_PContent" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:attribute name="tgtFrame" type="s:ST_String" use="optional"/>
        <xsd:attribute name="tooltip" type="s:ST_String" use="optional"/>
        <xsd:attribute name="docLocation" type="s:ST_String" use="optional"/>
        <xsd:attribute name="history" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="anchor" type="s:ST_String" use="optional"/>
        <xsd:attribute ref="r:id"/>
    </xsd:complexType>
```

```
    <xsd:complexType name="CT_FFData">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="name" type="CT_FFName"/>
            <xsd:element name="label" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="tabIndex" type="CT_UnsignedDecimalNumber" minOccurs="0"/>
            <xsd:element name="enabled" type="CT_OnOff"/>
            <xsd:element name="calcOnExit" type="CT_OnOff"/>
            <xsd:element name="entryMacro" type="CT_MacroName" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="exitMacro" type="CT_MacroName" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="helpText" type="CT_FFHelpText" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="statusText" type="CT_FFStatusText" minOccurs="0" maxOccurs="1"/>
            <xsd:choice>
                <xsd:element name="checkBox" type="CT_FFCheckBox"/>
                <xsd:element name="ddList" type="CT_FFDDList"/>
                <xsd:element name="textInput" type="CT_FFTextInput"/>
            </xsd:choice>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_FFHelpText">
        <xsd:attribute name="type" type="ST_InfoTextType"/>
        <xsd:attribute name="val" type="ST_FFHelpTextVal"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FFStatusText">
        <xsd:attribute name="type" type="ST_InfoTextType"/>
        <xsd:attribute name="val" type="ST_FFStatusTextVal"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FFCheckBox">
        <xsd:sequence>
            <xsd:choice>
                <xsd:element name="size" type="CT_HpsMeasure"/>
                <xsd:element name="sizeAuto" type="CT_OnOff"/>
            </xsd:choice>
            <xsd:element name="default" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="checked" type="CT_OnOff" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_FFDDList">
        <xsd:sequence>
            <xsd:element name="result" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="default" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="listEntry" type="CT_String" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_FFTextInput">
        <xsd:sequence>
            <xsd:element name="type" type="CT_FFTextType" minOccurs="0"/>
            <xsd:element name="default" type="CT_String" minOccurs="0"/>
            <xsd:element name="maxLength" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="format" type="CT_String" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_SectionMark">
        <xsd:restriction base="xsd:string">
```



```
            <xsd:enumeration value="nextPage"/>
            <xsd:enumeration value="nextColumn"/>
            <xsd:enumeration value="continuous"/>
            <xsd:enumeration value="evenPage"/>
            <xsd:enumeration value="oddPage"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SectType">
        <xsd:attribute name="val" type="ST_SectionMark"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PaperSource">
        <xsd:attribute name="first" type="ST_DecimalNumber"/>
        <xsd:attribute name="other" type="ST_DecimalNumber"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_NumberFormat">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="decimal"/>
            <xsd:enumeration value="upperRoman"/>
            <xsd:enumeration value="lowerRoman"/>
            <xsd:enumeration value="upperLetter"/>
            <xsd:enumeration value="lowerLetter"/>
            <xsd:enumeration value="ordinal"/>
            <xsd:enumeration value="cardinalText"/>
            <xsd:enumeration value="ordinalText"/>
            <xsd:enumeration value="hex"/>
            <xsd:enumeration value="chicago"/>
            <xsd:enumeration value="ideographDigital"/>
            <xsd:enumeration value="japaneseCounting"/>
            <xsd:enumeration value="aiueo"/>
            <xsd:enumeration value="iroha"/>
            <xsd:enumeration value="decimalFullWidth"/>
            <xsd:enumeration value="decimalHalfWidth"/>
            <xsd:enumeration value="japaneseLegal"/>
            <xsd:enumeration value="japaneseDigitalTenThousand"/>
            <xsd:enumeration value="decimalEnclosedCircle"/>
            <xsd:enumeration value="decimalFullWidth2"/>
            <xsd:enumeration value="aiueoFullWidth"/>
            <xsd:enumeration value="irohaFullWidth"/>
            <xsd:enumeration value="decimalZero"/>
            <xsd:enumeration value="bullet"/>
            <xsd:enumeration value="ganada"/>
            <xsd:enumeration value="chosung"/>
            <xsd:enumeration value="decimalEnclosedFullstop"/>
            <xsd:enumeration value="decimalEnclosedParen"/>
            <xsd:enumeration value="decimalEnclosedCircleChinese"/>
            <xsd:enumeration value="ideographEnclosedCircle"/>
            <xsd:enumeration value="ideographTraditional"/>
            <xsd:enumeration value="ideographZodiac"/>
            <xsd:enumeration value="ideographZodiacTraditional"/>
            <xsd:enumeration value="taiwaneseCounting"/>
            <xsd:enumeration value="ideographLegalTraditional"/>
            <xsd:enumeration value="taiwaneseCountingThousand"/>
            <xsd:enumeration value="taiwaneseDigital"/>
```

```
            <xsd:enumeration value="chineseCounting"/>
            <xsd:enumeration value="chineseLegalSimplified"/>
            <xsd:enumeration value="chineseCountingThousand"/>
            <xsd:enumeration value="koreanDigital"/>
            <xsd:enumeration value="koreanCounting"/>
            <xsd:enumeration value="koreanLegal"/>
            <xsd:enumeration value="koreanDigital2"/>
            <xsd:enumeration value="vietnameseCounting"/>
            <xsd:enumeration value="russianLower"/>
            <xsd:enumeration value="russianUpper"/>
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="numberInDash"/>
            <xsd:enumeration value="hebrew1"/>
            <xsd:enumeration value="hebrew2"/>
            <xsd:enumeration value="arabicAlpha"/>
            <xsd:enumeration value="arabicAbjad"/>
            <xsd:enumeration value="hindiVowels"/>
            <xsd:enumeration value="hindiConsonants"/>
            <xsd:enumeration value="hindiNumbers"/>
            <xsd:enumeration value="hindiCounting"/>
            <xsd:enumeration value="thaiLetters"/>
            <xsd:enumeration value="thaiNumbers"/>
            <xsd:enumeration value="thaiCounting"/>
            <xsd:enumeration value="bahtText"/>
            <xsd:enumeration value="dollarText"/>
            <xsd:enumeration value="custom"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PageOrientation">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="portrait"/>
            <xsd:enumeration value="landscape"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PageSz">
        <xsd:attribute name="w" type="s:ST_TwipsMeasure"/>
        <xsd:attribute name="h" type="s:ST_TwipsMeasure"/>
        <xsd:attribute name="orient" type="ST_PageOrientation" use="optional"/>
        <xsd:attribute name="code" type="ST_DecimalNumber" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PageMar">
        <xsd:attribute name="top" type="ST_SignedTwipsMeasure" use="required"/>
        <xsd:attribute name="right" type="s:ST_TwipsMeasure" use="required"/>
        <xsd:attribute name="bottom" type="ST_SignedTwipsMeasure" use="required"/>
        <xsd:attribute name="left" type="s:ST_TwipsMeasure" use="required"/>
        <xsd:attribute name="header" type="s:ST_TwipsMeasure" use="required"/>
        <xsd:attribute name="footer" type="s:ST_TwipsMeasure" use="required"/>
        <xsd:attribute name="gutter" type="s:ST_TwipsMeasure" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PageBorderZOrder">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="front"/>
            <xsd:enumeration value="back"/>
```



```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PageBorderDisplay">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="allPages"/>
            <xsd:enumeration value="firstPage"/>
            <xsd:enumeration value="notFirstPage"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PageBorderOffset">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="page"/>
            <xsd:enumeration value="text"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PageBorders">
        <xsd:sequence>
            <xsd:element name="top" type="CT_TopPageBorder" minOccurs="0"/>
            <xsd:element name="left" type="CT_PageBorder" minOccurs="0"/>
            <xsd:element name="bottom" type="CT_BottomPageBorder" minOccurs="0"/>
            <xsd:element name="right" type="CT_PageBorder" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="zOrder" type="ST_PageBorderZOrder" use="optional"/>
        <xsd:attribute name="display" type="ST_PageBorderDisplay" use="optional"/>
        <xsd:attribute name="offsetFrom" type="ST_PageBorderOffset" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PageBorder">
        <xsd:complexContent>
            <xsd:extension base="CT_Border">
                <xsd:attribute ref="r:id" use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_BottomPageBorder">
        <xsd:complexContent>
            <xsd:extension base="CT_PageBorder">
                <xsd:attribute ref="r:bottomLeft" use="optional"/>
                <xsd:attribute ref="r:bottomRight" use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_TopPageBorder">
        <xsd:complexContent>
            <xsd:extension base="CT_PageBorder">
                <xsd:attribute ref="r:topLeft" use="optional"/>
                <xsd:attribute ref="r:topRight" use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:simpleType name="ST_ChapterSep">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="hyphen"/>
            <xsd:enumeration value="period"/>
```

```
            <xsd:enumeration value="colon"/>
            <xsd:enumeration value="emDash"/>
            <xsd:enumeration value="enDash"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_LineNumberRestart">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="newPage"/>
            <xsd:enumeration value="newSection"/>
            <xsd:enumeration value="continuous"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_LineNumber">
        <xsd:attribute name="countBy" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="start" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="distance" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="restart" type="ST_LineNumberRestart" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PageNumber">
        <xsd:attribute name="fmt" type="ST_NumberFormat" use="optional"/>
        <xsd:attribute name="start" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="chapStyle" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="chapSep" type="ST_ChapterSep" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Column">
        <xsd:attribute name="w" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="space" type="s:ST_TwipsMeasure" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Columns">
        <xsd:sequence minOccurs="0">
            <xsd:element name="col" type="CT_Column" maxOccurs="45"/>
        </xsd:sequence>
        <xsd:attribute name="equalWidth" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="space" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="num" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="sep" type="s:ST_OnOff" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_VerticalJc">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="top"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="both"/>
            <xsd:enumeration value="bottom"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_VerticalJc">
        <xsd:attribute name="val" type="ST_VerticalJc" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DocGrid">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="default"/>
            <xsd:enumeration value="lines"/>
            <xsd:enumeration value="linesAndChars"/>
```



```
            <xsd:enumeration value="snapToChars"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DocGrid">
        <xsd:attribute name="type" type="ST_DocGrid"/>
        <xsd:attribute name="linePitch" type="ST_DecimalNumber"/>
        <xsd:attribute name="charSpace" type="ST_DecimalNumber"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_HdrFtr">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="even"/>
            <xsd:enumeration value="default"/>
            <xsd:enumeration value="first"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FtnEdn">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="normal"/>
            <xsd:enumeration value="separator"/>
            <xsd:enumeration value="continuationSeparator"/>
            <xsd:enumeration value="continuationNotice"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_HdrFtrRef">
        <xsd:complexContent>
            <xsd:extension base="CT_Rel">
                <xsd:attribute name="type" type="ST_HdrFtr" use="required"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:group name="EG_HdrFtrReferences">
        <xsd:choice>
            <xsd:element name="headerReference" type="CT_HdrFtrRef" minOccurs="0"/>
            <xsd:element name="footerReference" type="CT_HdrFtrRef" minOccurs="0"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_HdrFtr">
        <xsd:group ref="EG_BlockLevelElts" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:complexType>
    <xsd:group name="EG_SectPrContents">
        <xsd:sequence>
            <xsd:element name="footnotePr" type="CT_FtnProps" minOccurs="0"/>
            <xsd:element name="endnotePr" type="CT_EdnProps" minOccurs="0"/>
            <xsd:element name="type" type="CT_SectType" minOccurs="0"/>
            <xsd:element name="pgSz" type="CT_PageSz" minOccurs="0"/>
            <xsd:element name="pgMar" type="CT_PageMar" minOccurs="0"/>
            <xsd:element name="paperSrc" type="CT_PaperSource" minOccurs="0"/>
            <xsd:element name="pgBorders" type="CT_PageBorders" minOccurs="0"/>
            <xsd:element name="lnNumType" type="CT_LineNumber" minOccurs="0"/>
            <xsd:element name="pgNumType" type="CT_PageNumber" minOccurs="0"/>
            <xsd:element name="cols" type="CT_Columns" minOccurs="0"/>
            <xsd:element name="formProt" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="vAlign" type="CT_VerticalJc" minOccurs="0"/>
```

```
            <xsd:element name="noEndnote" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="titlePg" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="textDirection" type="CT_TextDirection" minOccurs="0"/>
            <xsd:element name="bidi" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="rtlGutter" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="docGrid" type="CT_DocGrid" minOccurs="0"/>
            <xsd:element name="printerSettings" type="CT_Rel" minOccurs="0"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:attributeGroup name="AG_SectPrAttributes">
        <xsd:attribute name="rsidRPr" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidDel" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidR" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidSect" type="ST_LongHexNumber"/>
    </xsd:attributeGroup>
    <xsd:complexType name="CT_SectPrBase">
        <xsd:sequence>
            <xsd:group ref="EG_SectPrContents" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_SectPrAttributes"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SectPr">
        <xsd:sequence>
            <xsd:group ref="EG_HdrFtrReferences" minOccurs="0" maxOccurs="6"/>
            <xsd:group ref="EG_SectPrContents" minOccurs="0"/>
            <xsd:element name="sectPrChange" type="CT_SectPrChange" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_SectPrAttributes"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_BrType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="page"/>
            <xsd:enumeration value="column"/>
            <xsd:enumeration value="textWrapping"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_BrClear">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="right"/>
            <xsd:enumeration value="all"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Br">
        <xsd:attribute name="type" type="ST_BrType" use="optional"/>
        <xsd:attribute name="clear" type="ST_BrClear" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PTabAlignment">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="right"/>
```



```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PTabRelativeTo">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="margin"/>
            <xsd:enumeration value="indent"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PTabLeader">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="dot"/>
            <xsd:enumeration value="hyphen"/>
            <xsd:enumeration value="underscore"/>
            <xsd:enumeration value="middleDot"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PTab">
        <xsd:attribute name="alignment" type="ST_PTabAlignment" use="required"/>
        <xsd:attribute name="relativeTo" type="ST_PTabRelativeTo" use="required"/>
        <xsd:attribute name="leader" type="ST_PTabLeader" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Sym">
        <xsd:attribute name="font" type="s:ST_String"/>
        <xsd:attribute name="char" type="ST_ShortHexNumber"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_ProofErr">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="spellStart"/>
            <xsd:enumeration value="spellEnd"/>
            <xsd:enumeration value="gramStart"/>
            <xsd:enumeration value="gramEnd"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_ProofErr">
        <xsd:attribute name="type" type="ST_ProofErr" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_EdGrp">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="everyone"/>
            <xsd:enumeration value="administrators"/>
            <xsd:enumeration value="contributors"/>
            <xsd:enumeration value="editors"/>
            <xsd:enumeration value="owners"/>
            <xsd:enumeration value="current"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Perm">
        <xsd:attribute name="id" type="s:ST_String" use="required"/>
        <xsd:attribute name="displacedByCustomXml" type="ST_DisplacedByCustomXml" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PermStart">
```

```
        <xsd:complexContent>
            <xsd:extension base="CT_Perm">
                <xsd:attribute name="edGrp" type="ST_EdGrp" use="optional"/>
                <xsd:attribute name="ed" type="s:ST_String" use="optional"/>
                <xsd:attribute name="colFirst" type="ST_DecimalNumber" use="optional"/>
                <xsd:attribute name="colLast" type="ST_DecimalNumber" use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_Text">
        <xsd:simpleContent>
            <xsd:extension base="s:ST_String">
                <xsd:attribute ref="xml:space" use="optional"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:group name="EG_RunInnerContent">
        <xsd:choice>
            <xsd:element name="br" type="CT_Br"/>
            <xsd:element name="t" type="CT_Text"/>
            <xsd:element name="contentPart" type="CT_Rel"/>
            <xsd:element name="delText" type="CT_Text"/>
            <xsd:element name="instrText" type="CT_Text"/>
            <xsd:element name="delInstrText" type="CT_Text"/>
            <xsd:element name="noBreakHyphen" type="CT_Empty"/>
            <xsd:element name="softHyphen" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="dayShort" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="monthShort" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="yearShort" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="dayLong" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="monthLong" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="yearLong" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="annotationRef" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="footnoteRef" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="endnoteRef" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="separator" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="continuationSeparator" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="sym" type="CT_Sym" minOccurs="0"/>
            <xsd:element name="pgNum" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="cr" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="tab" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="object" type="CT_Object"/>
            <xsd:element name="pict" type="CT_Picture"/>
            <xsd:element name="fldChar" type="CT_FldChar"/>
            <xsd:element name="ruby" type="CT_Ruby"/>
            <xsd:element name="footnoteReference" type="CT_FtnEdnRef"/>
            <xsd:element name="endnoteReference" type="CT_FtnEdnRef"/>
            <xsd:element name="commentReference" type="CT_Markup"/>
            <xsd:element name="drawing" type="CT_Drawing"/>
            <xsd:element name="ptab" type="CT_PTab" minOccurs="0"/>
            <xsd:element name="lastRenderedPageBreak" type="CT_Empty" minOccurs="0" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
```



```
    <xsd:complexType name="CT_R">
        <xsd:sequence>
            <xsd:group ref="EG_RPr" minOccurs="0"/>
            <xsd:group ref="EG_RunInnerContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="rsidRPr" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidDel" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidR" type="ST_LongHexNumber"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Hint">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="default"/>
            <xsd:enumeration value="eastAsia"/>
            <xsd:enumeration value="cs"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Theme">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="majorEastAsia"/>
            <xsd:enumeration value="majorBidi"/>
            <xsd:enumeration value="majorAscii"/>
            <xsd:enumeration value="majorHAnsi"/>
            <xsd:enumeration value="minorEastAsia"/>
            <xsd:enumeration value="minorBidi"/>
            <xsd:enumeration value="minorAscii"/>
            <xsd:enumeration value="minorHAnsi"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Fonts">
        <xsd:attribute name="hint" type="ST_Hint"/>
        <xsd:attribute name="ascii" type="s:ST_String"/>
        <xsd:attribute name="hAnsi" type="s:ST_String"/>
        <xsd:attribute name="eastAsia" type="s:ST_String"/>
        <xsd:attribute name="cs" type="s:ST_String"/>
        <xsd:attribute name="asciiTheme" type="ST_Theme"/>
        <xsd:attribute name="hAnsiTheme" type="ST_Theme"/>
        <xsd:attribute name="eastAsiaTheme" type="ST_Theme"/>
        <xsd:attribute name="cstheme" type="ST_Theme"/>
    </xsd:complexType>
    <xsd:group name="EG_RPrBase">
        <xsd:choice>
            <xsd:element name="rStyle" type="CT_String"/>
            <xsd:element name="rFonts" type="CT_Fonts"/>
            <xsd:element name="b" type="CT_OnOff"/>
            <xsd:element name="bCs" type="CT_OnOff"/>
            <xsd:element name="i" type="CT_OnOff"/>
            <xsd:element name="iCs" type="CT_OnOff"/>
            <xsd:element name="caps" type="CT_OnOff"/>
            <xsd:element name="smallCaps" type="CT_OnOff"/>
            <xsd:element name="strike" type="CT_OnOff"/>
            <xsd:element name="dstrike" type="CT_OnOff"/>
            <xsd:element name="outline" type="CT_OnOff"/>
            <xsd:element name="shadow" type="CT_OnOff"/>
```

```
            <xsd:element name="emboss" type="CT_OnOff"/>
            <xsd:element name="imprint" type="CT_OnOff"/>
            <xsd:element name="noProof" type="CT_OnOff"/>
            <xsd:element name="snapToGrid" type="CT_OnOff"/>
            <xsd:element name="vanish" type="CT_OnOff"/>
            <xsd:element name="webHidden" type="CT_OnOff"/>
            <xsd:element name="color" type="CT_Color"/>
            <xsd:element name="spacing" type="CT_SignedTwipsMeasure"/>
            <xsd:element name="w" type="CT_TextScale"/>
            <xsd:element name="kern" type="CT_HpsMeasure"/>
            <xsd:element name="position" type="CT_SignedHpsMeasure"/>
            <xsd:element name="sz" type="CT_HpsMeasure"/>
            <xsd:element name="szCs" type="CT_HpsMeasure"/>
            <xsd:element name="highlight" type="CT_Highlight"/>
            <xsd:element name="u" type="CT_Underline"/>
            <xsd:element name="effect" type="CT_TextEffect"/>
            <xsd:element name="bdr" type="CT_Border"/>
            <xsd:element name="shd" type="CT_Shd"/>
            <xsd:element name="fitText" type="CT_FitText"/>
            <xsd:element name="vertAlign" type="CT_VerticalAlignRun"/>
            <xsd:element name="rtl" type="CT_OnOff"/>
            <xsd:element name="cs" type="CT_OnOff"/>
            <xsd:element name="em" type="CT_Em"/>
            <xsd:element name="lang" type="CT_Language"/>
            <xsd:element name="eastAsianLayout" type="CT_EastAsianLayout"/>
            <xsd:element name="specVanish" type="CT_OnOff"/>
            <xsd:element name="oMath" type="CT_OnOff"/>
        </xsd:choice>
    </xsd:group>
    <xsd:group name="EG_RPrContent">
        <xsd:sequence>
            <xsd:group ref="EG_RPrBase" minOccurs="0"/>
            <xsd:element name="rPrChange" type="CT_RPrChange" minOccurs="0"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_RPr">
        <xsd:sequence>
            <xsd:group ref="EG_RPrContent" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_RPr">
        <xsd:sequence>
            <xsd:element name="rPr" type="CT_RPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:group name="EG_RPrMath">
        <xsd:choice>
            <xsd:group ref="EG_RPr"/>
            <xsd:element name="ins" type="CT_MathCtrlIns"/>
            <xsd:element name="del" type="CT_MathCtrlDel"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_MathCtrlIns">
```



```
      <xsd:complexContent>
        <xsd:extension base="CT_TrackChange">
               <xsd:choice minOccurs="0">
                      <xsd:element name="del" type="CT_RPrChange" minOccurs="1"/>
                      <xsd:element name="rPr" type="CT_RPr" minOccurs="1"/>
               </xsd:choice>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_MathCtrlDel">
      <xsd:complexContent>
        <xsd:extension base="CT_TrackChange">
               <xsd:choice minOccurs="0">
                      <xsd:element name="rPr" type="CT_RPr" minOccurs="1"/>
               </xsd:choice>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_RPrOriginal">
        <xsd:sequence>
            <xsd:group ref="EG_RPrBase" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ParaRPrOriginal">
        <xsd:sequence>
            <xsd:group ref="EG_ParaRPrTrackChanges" minOccurs="0"/>
            <xsd:group ref="EG_RPrBase" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ParaRPr">
        <xsd:sequence>
            <xsd:group ref="EG_ParaRPrTrackChanges" minOccurs="0"/>
            <xsd:group ref="EG_RPrBase" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rPrChange" type="CT_ParaRPrChange" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_ParaRPrTrackChanges">
        <xsd:sequence>
            <xsd:element name="ins" type="CT_TrackChange" minOccurs="0"/>
            <xsd:element name="del" type="CT_TrackChange" minOccurs="0"/>
            <xsd:element name="moveFrom" type="CT_TrackChange" minOccurs="0"/>
            <xsd:element name="moveTo" type="CT_TrackChange" minOccurs="0"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_AltChunk">
        <xsd:sequence>
            <xsd:element name="altChunkPr" type="CT_AltChunkPr" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute ref="r:id" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AltChunkPr">
        <xsd:sequence>
            <xsd:element name="matchSrc" type="CT_OnOff" minOccurs="0" maxOccurs="1"/>
```

```
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_RubyAlign">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="distributeLetter"/>
            <xsd:enumeration value="distributeSpace"/>
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="right"/>
            <xsd:enumeration value="rightVertical"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_RubyAlign">
        <xsd:attribute name="val" type="ST_RubyAlign" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RubyPr">
        <xsd:sequence>
            <xsd:element name="rubyAlign" type="CT_RubyAlign"/>
            <xsd:element name="hps" type="CT_HpsMeasure"/>
            <xsd:element name="hpsRaise" type="CT_HpsMeasure"/>
            <xsd:element name="hpsBaseText" type="CT_HpsMeasure"/>
            <xsd:element name="lid" type="CT_Lang"/>
            <xsd:element name="dirty" type="CT_OnOff" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_RubyContent">
        <xsd:choice>
            <xsd:element name="r" type="CT_R"/>
            <xsd:group ref="EG_RunLevelElts" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_RubyContent">
        <xsd:group ref="EG_RubyContent" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Ruby">
        <xsd:sequence>
            <xsd:element name="rubyPr" type="CT_RubyPr"/>
            <xsd:element name="rt" type="CT_RubyContent"/>
            <xsd:element name="rubyBase" type="CT_RubyContent"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_Lock">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="sdtLocked"/>
            <xsd:enumeration value="contentLocked"/>
            <xsd:enumeration value="unlocked"/>
            <xsd:enumeration value="sdtContentLocked"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Lock">
        <xsd:attribute name="val" type="ST_Lock"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtListItem">
```



```
        <xsd:attribute name="displayText" type="s:ST_String"/>
        <xsd:attribute name="value" type="s:ST_String"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SdtDateMappingType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="text"/>
            <xsd:enumeration value="date"/>
            <xsd:enumeration value="dateTime"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SdtDateMappingType">
        <xsd:attribute name="val" type="ST_SdtDateMappingType"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CalendarType">
        <xsd:attribute name="val" type="s:ST_CalendarType"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtDate">
        <xsd:sequence>
            <xsd:element name="dateFormat" type="CT_String" minOccurs="0"/>
            <xsd:element name="lid" type="CT_Lang" minOccurs="0"/>
            <xsd:element name="storeMappedDataAs" type="CT_SdtDateMappingType" minOccurs="0"/>
            <xsd:element name="calendar" type="CT_CalendarType" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="fullDate" type="ST_DateTime" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtComboBox">
        <xsd:sequence>
            <xsd:element name="listItem" type="CT_SdtListItem" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="lastValue" type="s:ST_String" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtDocPart">
        <xsd:sequence>
            <xsd:element name="docPartGallery" type="CT_String" minOccurs="0"/>
            <xsd:element name="docPartCategory" type="CT_String" minOccurs="0"/>
            <xsd:element name="docPartUnique" type="CT_OnOff" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtDropDownList">
        <xsd:sequence>
            <xsd:element name="listItem" type="CT_SdtListItem" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="lastValue" type="s:ST_String" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Placeholder">
        <xsd:sequence>
            <xsd:element name="docPart" type="CT_String"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtText">
        <xsd:attribute name="multiLine" type="s:ST_OnOff"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DataBinding">
```

```
        <xsd:attribute name="prefixMappings" type="s:ST_String"/>
        <xsd:attribute name="xpath" type="s:ST_String" use="required"/>
        <xsd:attribute name="storeItemID" type="s:ST_String" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtPr">
        <xsd:sequence>
            <xsd:element name="rPr" type="CT_RPr" minOccurs="0"/>
            <xsd:element name="alias" type="CT_String" minOccurs="0"/>
            <xsd:element name="tag" type="CT_String" minOccurs="0"/>
            <xsd:element name="id" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="lock" type="CT_Lock" minOccurs="0"/>
            <xsd:element name="placeholder" type="CT_Placeholder" minOccurs="0"/>
            <xsd:element name="temporary" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="showingPlcHdr" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="dataBinding" type="CT_DataBinding" minOccurs="0"/>
            <xsd:element name="label" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="tabIndex" type="CT_UnsignedDecimalNumber" minOccurs="0"/>
            <xsd:choice minOccurs="0" maxOccurs="1">
                <xsd:element name="equation" type="CT_Empty"/>
                <xsd:element name="comboBox" type="CT_SdtComboBox"/>
                <xsd:element name="date" type="CT_SdtDate"/>
                <xsd:element name="docPartObj" type="CT_SdtDocPart"/>
                <xsd:element name="docPartList" type="CT_SdtDocPart"/>
                <xsd:element name="dropDownList" type="CT_SdtDropDownList"/>
                <xsd:element name="picture" type="CT_Empty"/>
                <xsd:element name="richText" type="CT_Empty"/>
                <xsd:element name="text" type="CT_SdtText"/>
                <xsd:element name="citation" type="CT_Empty"/>
                <xsd:element name="group" type="CT_Empty"/>
                <xsd:element name="bibliography" type="CT_Empty"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtEndPr">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="rPr" type="CT_RPr" minOccurs="0"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:group name="EG_ContentRunContent">
        <xsd:choice>
            <xsd:element name="customXml" type="CT_CustomXmlRun"/>
            <xsd:element name="smartTag" type="CT_SmartTagRun"/>
            <xsd:element name="sdt" type="CT_SdtRun"/>
            <xsd:element name="dir" type="CT_DirContentRun"/>
            <xsd:element name="bdo" type="CT_BdoContentRun"/>
            <xsd:element name="r" type="CT_R"/>
            <xsd:group ref="EG_RunLevelElts" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_DirContentRun">
        <xsd:group ref="EG_PContent" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:attribute name="val" type="ST_Direction" use="optional"/>
    </xsd:complexType>
```



```
    <xsd:complexType name="CT_BdoContentRun">
        <xsd:group ref="EG_PContent" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:attribute name="val" type="ST_Direction" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Direction">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="ltr"/>
            <xsd:enumeration value="rtl"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SdtContentRun">
        <xsd:group ref="EG_PContent" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:complexType>
    <xsd:group name="EG_ContentBlockContent">
        <xsd:choice>
            <xsd:element name="customXml" type="CT_CustomXmlBlock"/>
            <xsd:element name="sdt" type="CT_SdtBlock"/>
            <xsd:element name="p" type="CT_P" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="tbl" type="CT_Tbl" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:group ref="EG_RunLevelElts" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_SdtContentBlock">
        <xsd:group ref="EG_ContentBlockContent" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:complexType>
    <xsd:group name="EG_ContentRowContent">
        <xsd:choice>
            <xsd:element name="tr" type="CT_Row" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="customXml" type="CT_CustomXmlRow"/>
            <xsd:element name="sdt" type="CT_SdtRow"/>
            <xsd:group ref="EG_RunLevelElts" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_SdtContentRow">
        <xsd:group ref="EG_ContentRowContent" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:complexType>
    <xsd:group name="EG_ContentCellContent">
        <xsd:choice>
            <xsd:element name="tc" type="CT_Tc" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="customXml" type="CT_CustomXmlCell"/>
            <xsd:element name="sdt" type="CT_SdtCell"/>
            <xsd:group ref="EG_RunLevelElts" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_SdtContentCell">
        <xsd:group ref="EG_ContentCellContent" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtBlock">
        <xsd:sequence>
            <xsd:element name="sdtPr" type="CT_SdtPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sdtEndPr" type="CT_SdtEndPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sdtContent" type="CT_SdtContentBlock" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
```

```
    </xsd:complexType>
    <xsd:complexType name="CT_SdtRun">
        <xsd:sequence>
            <xsd:element name="sdtPr" type="CT_SdtPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sdtEndPr" type="CT_SdtEndPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sdtContent" type="CT_SdtContentRun" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtCell">
        <xsd:sequence>
            <xsd:element name="sdtPr" type="CT_SdtPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sdtEndPr" type="CT_SdtEndPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sdtContent" type="CT_SdtContentCell" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtRow">
        <xsd:sequence>
            <xsd:element name="sdtPr" type="CT_SdtPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sdtEndPr" type="CT_SdtEndPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sdtContent" type="CT_SdtContentRow" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Attr">
        <xsd:attribute name="uri" type="s:ST_String"/>
        <xsd:attribute name="name" type="s:ST_String" use="required"/>
        <xsd:attribute name="val" type="s:ST_String" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomXmlRun">
        <xsd:sequence>
            <xsd:element name="customXmlPr" type="CT_CustomXmlPr" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_PContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="uri" type="s:ST_String"/>
        <xsd:attribute name="element" type="s:ST_XmlName" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SmartTagRun">
        <xsd:sequence>
            <xsd:element name="smartTagPr" type="CT_SmartTagPr" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_PContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="uri" type="s:ST_String"/>
        <xsd:attribute name="element" type="s:ST_XmlName" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomXmlBlock">
        <xsd:sequence>
            <xsd:element name="customXmlPr" type="CT_CustomXmlPr" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_ContentBlockContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="uri" type="s:ST_String"/>
        <xsd:attribute name="element" type="s:ST_XmlName" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomXmlPr">
        <xsd:sequence>
```



```
            <xsd:element name="placeholder" type="CT_String" minOccurs="0"/>
            <xsd:element name="attr" type="CT_Attr" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomXmlRow">
        <xsd:sequence>
            <xsd:element name="customXmlPr" type="CT_CustomXmlPr" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_ContentRowContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="uri" type="s:ST_String"/>
        <xsd:attribute name="element" type="s:ST_XmlName" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomXmlCell">
        <xsd:sequence>
            <xsd:element name="customXmlPr" type="CT_CustomXmlPr" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_ContentCellContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="uri" type="s:ST_String"/>
        <xsd:attribute name="element" type="s:ST_XmlName" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SmartTagPr">
        <xsd:sequence>
            <xsd:element name="attr" type="CT_Attr" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_PContent">
        <xsd:choice>
            <xsd:group ref="EG_ContentRunContent" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="fldSimple" type="CT_SimpleField" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="hyperlink" type="CT_Hyperlink"/>
            <xsd:element name="subDoc" type="CT_Rel"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_P">
        <xsd:sequence>
            <xsd:element name="pPr" type="CT_PPr" minOccurs="0"/>
            <xsd:group ref="EG_PContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="rsidRPr" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidR" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidDel" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidP" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidRDefault" type="ST_LongHexNumber"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TblWidth">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="nil"/>
            <xsd:enumeration value="pct"/>
            <xsd:enumeration value="dxa"/>
            <xsd:enumeration value="auto"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Height">
```

```
        <xsd:attribute name="val" type="s:ST_TwipsMeasure"/>
        <xsd:attribute name="hRule" type="ST_HeightRule"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_MeasurementOrPercent">
        <xsd:union memberTypes="ST_DecimalNumberOrPercent s:ST_UniversalMeasure"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_TblWidth">
        <xsd:attribute name="w" type="ST_MeasurementOrPercent"/>
        <xsd:attribute name="type" type="ST_TblWidth"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TblGridCol">
        <xsd:attribute name="w" type="s:ST_TwipsMeasure"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TblGridBase">
        <xsd:sequence>
            <xsd:element name="gridCol" type="CT_TblGridCol" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TblGrid">
        <xsd:complexContent>
            <xsd:extension base="CT_TblGridBase">
                <xsd:sequence>
                    <xsd:element name="tblGridChange" type="CT_TblGridChange" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_TcBorders">
        <xsd:sequence>
            <xsd:element name="top" type="CT_Border" minOccurs="0"/>
            <xsd:element name="start" type="CT_Border" minOccurs="0"/>
            <xsd:element name="left" type="CT_Border" minOccurs="0"/>
            <xsd:element name="bottom" type="CT_Border" minOccurs="0"/>
            <xsd:element name="end" type="CT_Border" minOccurs="0"/>
            <xsd:element name="right" type="CT_Border" minOccurs="0"/>
            <xsd:element name="insideH" type="CT_Border" minOccurs="0"/>
            <xsd:element name="insideV" type="CT_Border" minOccurs="0"/>
            <xsd:element name="tl2br" type="CT_Border" minOccurs="0"/>
            <xsd:element name="tr2bl" type="CT_Border" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TcMar">
        <xsd:sequence>
            <xsd:element name="top" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="start" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="left" type="CT_TblWidth" minOccurs="0"/>
            <xsd:element name="bottom" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="end" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="right" type="CT_TblWidth" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_Merge">
        <xsd:restriction base="xsd:string">
```



```
            <xsd:enumeration value="continue"/>
            <xsd:enumeration value="restart"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_VMerge">
        <xsd:attribute name="val" type="ST_Merge"/>
    </xsd:complexType>
    <xsd:complexType name="CT_HMerge">
        <xsd:attribute name="val" type="ST_Merge"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TcPrBase">
        <xsd:sequence>
            <xsd:element name="cnfStyle" type="CT_Cnf" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tcW" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="gridSpan" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="hMerge" type="CT_HMerge" minOccurs="0"/>
            <xsd:element name="vMerge" type="CT_VMerge" minOccurs="0"/>
            <xsd:element name="tcBorders" type="CT_TcBorders" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="shd" type="CT_Shd" minOccurs="0"/>
            <xsd:element name="noWrap" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="tcMar" type="CT_TcMar" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="textDirection" type="CT_TextDirection" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tcFitText" type="CT_OnOff" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="vAlign" type="CT_VerticalJc" minOccurs="0"/>
            <xsd:element name="hideMark" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="headers" type="CT_Headers" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TcPr">
        <xsd:complexContent>
            <xsd:extension base="CT_TcPrInner">
                <xsd:sequence>
                    <xsd:element name="tcPrChange" type="CT_TcPrChange" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_TcPrInner">
        <xsd:complexContent>
            <xsd:extension base="CT_TcPrBase">
                <xsd:sequence>
                    <xsd:group ref="EG_CellMarkupElements" minOccurs="0" maxOccurs="1"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_Tc">
        <xsd:sequence>
            <xsd:element name="tcPr" type="CT_TcPr" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_BlockLevelElts" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="s:ST_String" use="optional"/>
    </xsd:complexType>
```

```
    <xsd:simpleType name="ST_Cnf">
        <xsd:restriction base="xsd:string">
            <xsd:length value="12"/>
            <xsd:pattern value="[01]*"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Cnf">
        <xsd:attribute name="val" type="ST_Cnf"/>
        <xsd:attribute name="firstRow" type="s:ST_OnOff"/>
        <xsd:attribute name="lastRow" type="s:ST_OnOff"/>
        <xsd:attribute name="firstColumn" type="s:ST_OnOff"/>
        <xsd:attribute name="lastColumn" type="s:ST_OnOff"/>
        <xsd:attribute name="oddVBand" type="s:ST_OnOff"/>
        <xsd:attribute name="evenVBand" type="s:ST_OnOff"/>
        <xsd:attribute name="oddHBand" type="s:ST_OnOff"/>
        <xsd:attribute name="evenHBand" type="s:ST_OnOff"/>
        <xsd:attribute name="firstRowFirstColumn" type="s:ST_OnOff"/>
        <xsd:attribute name="firstRowLastColumn" type="s:ST_OnOff"/>
        <xsd:attribute name="lastRowFirstColumn" type="s:ST_OnOff"/>
        <xsd:attribute name="lastRowLastColumn" type="s:ST_OnOff"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Headers">
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="header" type="CT_String"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TrPrBase">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="cnfStyle" type="CT_Cnf" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="divId" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="gridBefore" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="gridAfter" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="wBefore" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="wAfter" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cantSplit" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="trHeight" type="CT_Height" minOccurs="0"/>
            <xsd:element name="tblHeader" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="tblCellSpacing" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="jc" type="CT_JcTable" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="hidden" type="CT_OnOff" minOccurs="0"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_TrPr">
        <xsd:complexContent>
            <xsd:extension base="CT_TrPrBase">
                <xsd:sequence>
                    <xsd:element name="ins" type="CT_TrackChange" minOccurs="0"/>
                    <xsd:element name="del" type="CT_TrackChange" minOccurs="0"/>
                    <xsd:element name="trPrChange" type="CT_TrPrChange" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
```



```
    <xsd:complexType name="CT_Row">
        <xsd:sequence>
            <xsd:element name="tblPrEx" type="CT_TblPrEx" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="trPr" type="CT_TrPr" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_ContentCellContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="rsidRPr" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidR" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidDel" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidTr" type="ST_LongHexNumber"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TblLayoutType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="fixed"/>
            <xsd:enumeration value="autofit"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TblLayoutType">
        <xsd:attribute name="type" type="ST_TblLayoutType"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TblOverlap">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="never"/>
            <xsd:enumeration value="overlap"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TblOverlap">
        <xsd:attribute name="val" type="ST_TblOverlap" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TblPPr">
        <xsd:attribute name="leftFromText" type="s:ST_TwipsMeasure"/>
        <xsd:attribute name="rightFromText" type="s:ST_TwipsMeasure"/>
        <xsd:attribute name="topFromText" type="s:ST_TwipsMeasure"/>
        <xsd:attribute name="bottomFromText" type="s:ST_TwipsMeasure"/>
        <xsd:attribute name="vertAnchor" type="ST_VAnchor"/>
        <xsd:attribute name="horzAnchor" type="ST_HAnchor"/>
        <xsd:attribute name="tblpXSpec" type="s:ST_XAlign"/>
        <xsd:attribute name="tblpX" type="ST_SignedTwipsMeasure"/>
        <xsd:attribute name="tblpYSpec" type="s:ST_YAlign"/>
        <xsd:attribute name="tblpY" type="ST_SignedTwipsMeasure"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TblCellMar">
        <xsd:sequence>
            <xsd:element name="top" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="start" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="left" type="CT_TblWidth" minOccurs="0"/>
            <xsd:element name="bottom" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="end" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="right" type="CT_TblWidth" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TblBorders">
        <xsd:sequence>
```

```
            <xsd:element name="top" type="CT_Border" minOccurs="0"/>
            <xsd:element name="start" type="CT_Border" minOccurs="0"/>
            <xsd:element name="left" type="CT_Border" minOccurs="0"/>
            <xsd:element name="bottom" type="CT_Border" minOccurs="0"/>
            <xsd:element name="end" type="CT_Border" minOccurs="0"/>
            <xsd:element name="right" type="CT_Border" minOccurs="0"/>
            <xsd:element name="insideH" type="CT_Border" minOccurs="0"/>
            <xsd:element name="insideV" type="CT_Border" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TblPrBase">
        <xsd:sequence>
            <xsd:element name="tblStyle" type="CT_String" minOccurs="0"/>
            <xsd:element name="tblpPr" type="CT_TblPPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblOverlap" type="CT_TblOverlap" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bidiVisual" type="CT_OnOff" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblStyleRowBandSize" type="CT_DecimalNumber" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="tblStyleColBandSize" type="CT_DecimalNumber" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="tblW" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="jc" type="CT_JcTable" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblCellSpacing" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblInd" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblBorders" type="CT_TblBorders" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="shd" type="CT_Shd" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblLayout" type="CT_TblLayoutType" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblCellMar" type="CT_TblCellMar" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblLook" type="CT_TblLook" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblCaption" type="CT_String" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblDescription" type="CT_String" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TblPr">
        <xsd:complexContent>
            <xsd:extension base="CT_TblPrBase">
                <xsd:sequence>
                    <xsd:element name="tblPrChange" type="CT_TblPrChange" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_TblPrExBase">
        <xsd:sequence>
            <xsd:element name="tblW" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="jc" type="CT_JcTable" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblCellSpacing" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblInd" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblBorders" type="CT_TblBorders" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="shd" type="CT_Shd" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblLayout" type="CT_TblLayoutType" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblCellMar" type="CT_TblCellMar" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblLook" type="CT_TblLook" minOccurs="0" maxOccurs="1"/>
```



```
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TblPrEx">
        <xsd:complexContent>
            <xsd:extension base="CT_TblPrExBase">
                <xsd:sequence>
                    <xsd:element name="tblPrExChange" type="CT_TblPrExChange" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_Tbl">
        <xsd:sequence>
            <xsd:group ref="EG_RangeMarkupElements" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="tblPr" type="CT_TblPr"/>
            <xsd:element name="tblGrid" type="CT_TblGrid"/>
            <xsd:group ref="EG_ContentRowContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TblLook">
        <xsd:attribute name="firstRow" type="s:ST_OnOff"/>
        <xsd:attribute name="lastRow" type="s:ST_OnOff"/>
        <xsd:attribute name="firstColumn" type="s:ST_OnOff"/>
        <xsd:attribute name="lastColumn" type="s:ST_OnOff"/>
        <xsd:attribute name="noHBand" type="s:ST_OnOff"/>
        <xsd:attribute name="noVBand" type="s:ST_OnOff"/>
        <xsd:attribute name="val" type="ST_ShortHexNumber"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_FtnPos">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="pageBottom"/>
            <xsd:enumeration value="beneathText"/>
            <xsd:enumeration value="sectEnd"/>
            <xsd:enumeration value="docEnd"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_FtnPos">
        <xsd:attribute name="val" type="ST_FtnPos" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_EdnPos">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="sectEnd"/>
            <xsd:enumeration value="docEnd"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_EdnPos">
        <xsd:attribute name="val" type="ST_EdnPos" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NumFmt">
        <xsd:attribute name="val" type="ST_NumberFormat" use="required"/>
        <xsd:attribute name="format" type="s:ST_String" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_RestartNumber">
```

```
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="continuous"/>
            <xsd:enumeration value="eachSect"/>
            <xsd:enumeration value="eachPage"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_NumRestart">
        <xsd:attribute name="val" type="ST_RestartNumber" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FtnEdnRef">
        <xsd:attribute name="customMarkFollows" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="id" use="required" type="ST_DecimalNumber"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FtnEdnSepRef">
        <xsd:attribute name="id" type="ST_DecimalNumber" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FtnEdn">
        <xsd:sequence>
            <xsd:group ref="EG_BlockLevelElts" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="type" type="ST_FtnEdn" use="optional"/>
        <xsd:attribute name="id" type="ST_DecimalNumber" use="required"/>
    </xsd:complexType>
    <xsd:group name="EG_FtnEdnNumProps">
        <xsd:sequence>
            <xsd:element name="numStart" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="numRestart" type="CT_NumRestart" minOccurs="0"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_FtnProps">
        <xsd:sequence>
            <xsd:element name="pos" type="CT_FtnPos" minOccurs="0"/>
            <xsd:element name="numFmt" type="CT_NumFmt" minOccurs="0"/>
            <xsd:group ref="EG_FtnEdnNumProps" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_EdnProps">
        <xsd:sequence>
            <xsd:element name="pos" type="CT_EdnPos" minOccurs="0"/>
            <xsd:element name="numFmt" type="CT_NumFmt" minOccurs="0"/>
            <xsd:group ref="EG_FtnEdnNumProps" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_FtnDocProps">
        <xsd:complexContent>
            <xsd:extension base="CT_FtnProps">
                <xsd:sequence>
                    <xsd:element name="footnote" type="CT_FtnEdnSepRef" minOccurs="0" maxOccurs="3"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_EdnDocProps">
```



```
        <xsd:complexContent>
            <xsd:extension base="CT_EdnProps">
                <xsd:sequence>
                    <xsd:element name="endnote" type="CT_FtnEdnSepRef" minOccurs="0" maxOccurs="3"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_RecipientData">
        <xsd:sequence>
            <xsd:element name="active" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="column" type="CT_DecimalNumber" minOccurs="1"/>
            <xsd:element name="uniqueTag" type="CT_Base64Binary" minOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Base64Binary">
        <xsd:attribute name="val" type="xsd:base64Binary" use="required">
        </xsd:attribute>
    </xsd:complexType>
    <xsd:complexType name="CT_Recipients">
        <xsd:sequence>
            <xsd:element name="recipientData" type="CT_RecipientData" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="recipients" type="CT_Recipients"/>
    <xsd:complexType name="CT_OdsoFieldMapData">
        <xsd:sequence>
            <xsd:element name="type" type="CT_MailMergeOdsoFMDFieldType" minOccurs="0"/>
            <xsd:element name="name" type="CT_String" minOccurs="0"/>
            <xsd:element name="mappedName" type="CT_String" minOccurs="0"/>
            <xsd:element name="column" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="lid" type="CT_Lang" minOccurs="0"/>
            <xsd:element name="dynamicAddress" type="CT_OnOff" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_MailMergeSourceType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="database"/>
            <xsd:enumeration value="addressBook"/>
            <xsd:enumeration value="document1"/>
            <xsd:enumeration value="document2"/>
            <xsd:enumeration value="text"/>
            <xsd:enumeration value="email"/>
            <xsd:enumeration value="native"/>
            <xsd:enumeration value="legacy"/>
            <xsd:enumeration value="master"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_MailMergeSourceType">
        <xsd:attribute name="val" use="required" type="ST_MailMergeSourceType"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Odso">
```

```
        <xsd:sequence>
            <xsd:element name="udl" type="CT_String" minOccurs="0"/>
            <xsd:element name="table" type="CT_String" minOccurs="0"/>
            <xsd:element name="src" type="CT_Rel" minOccurs="0"/>
            <xsd:element name="colDelim" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="type" type="CT_MailMergeSourceType" minOccurs="0"/>
            <xsd:element name="fHdr" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="fieldMapData" type="CT_OdsoFieldMapData" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="recipientData" type="CT_Rel" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_MailMerge">
        <xsd:sequence>
            <xsd:element name="mainDocumentType" type="CT_MailMergeDocType" minOccurs="1"/>
            <xsd:element name="linkToQuery" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="dataType" type="CT_MailMergeDataType" minOccurs="1"/>
            <xsd:element name="connectString" type="CT_String" minOccurs="0"/>
            <xsd:element name="query" type="CT_String" minOccurs="0"/>
            <xsd:element name="dataSource" type="CT_Rel" minOccurs="0"/>
            <xsd:element name="headerSource" type="CT_Rel" minOccurs="0"/>
            <xsd:element name="doNotSuppressBlankLines" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="destination" type="CT_MailMergeDest" minOccurs="0"/>
            <xsd:element name="addressFieldName" type="CT_String" minOccurs="0"/>
            <xsd:element name="mailSubject" type="CT_String" minOccurs="0"/>
            <xsd:element name="mailAsAttachment" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="viewMergedData" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="activeRecord" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="checkErrors" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="odso" type="CT_Odso" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_TargetScreenSz">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="544x376"/>
            <xsd:enumeration value="640x480"/>
            <xsd:enumeration value="720x512"/>
            <xsd:enumeration value="800x600"/>
            <xsd:enumeration value="1024x768"/>
            <xsd:enumeration value="1152x882"/>
            <xsd:enumeration value="1152x900"/>
            <xsd:enumeration value="1280x1024"/>
            <xsd:enumeration value="1600x1200"/>
            <xsd:enumeration value="1800x1440"/>
            <xsd:enumeration value="1920x1200"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TargetScreenSz">
        <xsd:attribute name="val" type="ST_TargetScreenSz" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Compat">
        <xsd:sequence>
            <xsd:element name="useSingleBorderforContiguousCells" type="CT_OnOff" minOccurs="0"/>
```



```
            <xsd:element name="wpJustification" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="noTabHangInd" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="noLeading" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="spaceForUL" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="noColumnBalance" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="balanceSingleByteDoubleByteWidth" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="noExtraLineSpacing" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotLeaveBackslashAlone" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="ulTrailSpace" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotExpandShiftReturn" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="spacingInWholePoints" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="lineWrapLikeWord6" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="printBodyTextBeforeHeader" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="printColBlack" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="wpSpaceWidth" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="showBreaksInFrames" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="subFontBySize" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="suppressBottomSpacing" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="suppressTopSpacing" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="suppressSpacingAtTopOfPage" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="suppressTopSpacingWP" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="suppressSpBfAfterPgBrk" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="swapBordersFacingPages" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="convMailMergeEsc" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="truncateFontHeightsLikeWP6" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="mwSmallCaps" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="usePrinterMetrics" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotSuppressParagraphBorders" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="wrapTrailSpaces" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="footnoteLayoutLikeWW8" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="shapeLayoutLikeWW8" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="alignTablesRowByRow" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="forgetLastTabAlignment" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="adjustLineHeightInTable" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="autoSpaceLikeWord95" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="noSpaceRaiseLower" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotUseHTMLParagraphAutoSpacing" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="layoutRawTableWidth" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="layoutTableRowsApart" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="useWord97LineBreakRules" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotBreakWrappedTables" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotSnapToGridInCell" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="selectFldWithFirstOrLastChar" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="applyBreakingRules" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotWrapTextWithPunct" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotUseEastAsianBreakRules" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="useWord2002TableStyleRules" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="growAutofit" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="useFELayout" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="useNormalStyleForList" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotUseIndentAsNumberingTabStop" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="useAltKinsokuLineBreakRules" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="allowSpaceOfSameStyleInTable" type="CT_OnOff" minOccurs="0"/>
```

```
            <xsd:element name="doNotSuppressIndentation" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotAutofitConstrainedTables" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="autofitToFirstFixedWidthCell" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="underlineTabInNumList" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="displayHangulFixedWidth" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="splitPgBreakAndParaMark" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotVertAlignCellWithSp" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotBreakConstrainedForcedTable" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotVertAlignInTxbx" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="useAnsiKerningPairs" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="cachedColBalance" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="compatSetting" type="CT_CompatSetting" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CompatSetting">
        <xsd:attribute name="name" type="s:ST_String"/>
        <xsd:attribute name="uri" type="s:ST_String"/>
        <xsd:attribute name="val" type="s:ST_String"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DocVar">
        <xsd:attribute name="name" type="s:ST_String" use="required"/>
        <xsd:attribute name="val" type="s:ST_String" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DocVars">
        <xsd:sequence>
            <xsd:element name="docVar" type="CT_DocVar" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DocRsids">
        <xsd:sequence>
            <xsd:element name="rsidRoot" type="CT_LongHexNumber" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="rsid" type="CT_LongHexNumber" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_CharacterSpacing">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="doNotCompress"/>
            <xsd:enumeration value="compressPunctuation"/>
            <xsd:enumeration value="compressPunctuationAndJapaneseKana"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_CharacterSpacing">
        <xsd:attribute name="val" type="ST_CharacterSpacing" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SaveThroughXslt">
        <xsd:attribute ref="r:id" use="optional"/>
        <xsd:attribute name="solutionID" type="s:ST_String" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RPrDefault">
        <xsd:sequence>
            <xsd:element name="rPr" type="CT_RPr" minOccurs="0"/>
        </xsd:sequence>
```



```
    </xsd:complexType>
    <xsd:complexType name="CT_PPrDefault">
        <xsd:sequence>
            <xsd:element name="pPr" type="CT_PPrGeneral" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DocDefaults">
        <xsd:sequence>
            <xsd:element name="rPrDefault" type="CT_RPrDefault" minOccurs="0"/>
            <xsd:element name="pPrDefault" type="CT_PPrDefault" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_WmlColorSchemeIndex">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="dark1"/>
            <xsd:enumeration value="light1"/>
            <xsd:enumeration value="dark2"/>
            <xsd:enumeration value="light2"/>
            <xsd:enumeration value="accent1"/>
            <xsd:enumeration value="accent2"/>
            <xsd:enumeration value="accent3"/>
            <xsd:enumeration value="accent4"/>
            <xsd:enumeration value="accent5"/>
            <xsd:enumeration value="accent6"/>
            <xsd:enumeration value="hyperlink"/>
            <xsd:enumeration value="followedHyperlink"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_ColorSchemeMapping">
        <xsd:attribute name="bg1" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="t1" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="bg2" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="t2" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="accent1" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="accent2" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="accent3" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="accent4" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="accent5" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="accent6" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="hyperlink" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="followedHyperlink" type="ST_WmlColorSchemeIndex"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ReadingModeInkLockDown">
        <xsd:attribute name="actualPg" type="s:ST_OnOff" use="required"/>
        <xsd:attribute name="w" type="ST_PixelsMeasure" use="required"/>
        <xsd:attribute name="h" type="ST_PixelsMeasure" use="required"/>
        <xsd:attribute name="fontSz" type="ST_DecimalNumberOrPercent" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WriteProtection">
        <xsd:attribute name="recommended" type="s:ST_OnOff" use="optional"/>
        <xsd:attributeGroup ref="AG_Password"/>
        <xsd:attributeGroup ref="AG_TransitionalPassword"/>
    </xsd:complexType>
```

```
    <xsd:complexType name="CT_Settings">
        <xsd:sequence>
            <xsd:element name="writeProtection" type="CT_WriteProtection" minOccurs="0"/>
            <xsd:element name="view" type="CT_View" minOccurs="0"/>
            <xsd:element name="zoom" type="CT_Zoom" minOccurs="0"/>
            <xsd:element name="removePersonalInformation" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="removeDateAndTime" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotDisplayPageBoundaries" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="displayBackgroundShape" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="printPostScriptOverText" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="printFractionalCharacterWidth" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="printFormsData" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="embedTrueTypeFonts" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="embedSystemFonts" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="saveSubsetFonts" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="saveFormsData" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="mirrorMargins" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="alignBordersAndEdges" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="bordersDoNotSurroundHeader" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="bordersDoNotSurroundFooter" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="gutterAtTop" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="hideSpellingErrors" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="hideGrammaticalErrors" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="activeWritingStyle" type="CT_WritingStyle" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="proofState" type="CT_Proof" minOccurs="0"/>
            <xsd:element name="formsDesign" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="attachedTemplate" type="CT_Rel" minOccurs="0"/>
            <xsd:element name="linkStyles" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="stylePaneFormatFilter" type="CT_StylePaneFilter" minOccurs="0"/>
            <xsd:element name="stylePaneSortMethod" type="CT_StyleSort" minOccurs="0"/>
            <xsd:element name="documentType" type="CT_DocType" minOccurs="0"/>
            <xsd:element name="mailMerge" type="CT_MailMerge" minOccurs="0"/>
            <xsd:element name="revisionView" type="CT_TrackChangesView" minOccurs="0"/>
            <xsd:element name="trackRevisions" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotTrackMoves" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotTrackFormatting" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="documentProtection" type="CT_DocProtect" minOccurs="0"/>
            <xsd:element name="autoFormatOverride" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="styleLockTheme" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="styleLockQFSet" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="defaultTabStop" type="CT_TwipsMeasure" minOccurs="0"/>
            <xsd:element name="autoHyphenation" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="consecutiveHyphenLimit" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="hyphenationZone" type="CT_TwipsMeasure" minOccurs="0"/>
            <xsd:element name="doNotHyphenateCaps" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="showEnvelope" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="summaryLength" type="CT_DecimalNumberOrPrecent" minOccurs="0"/>
            <xsd:element name="clickAndTypeStyle" type="CT_String" minOccurs="0"/>
            <xsd:element name="defaultTableStyle" type="CT_String" minOccurs="0"/>
            <xsd:element name="evenAndOddHeaders" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="bookFoldRevPrinting" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="bookFoldPrinting" type="CT_OnOff" minOccurs="0"/>
```



```
            <xsd:element name="bookFoldPrintingSheets" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="drawingGridHorizontalSpacing" type="CT_TwipsMeasure" minOccurs="0"/>
            <xsd:element name="drawingGridVerticalSpacing" type="CT_TwipsMeasure" minOccurs="0"/>
            <xsd:element name="displayHorizontalDrawingGridEvery" type="CT_DecimalNumber"
              minOccurs="0"/>
            <xsd:element name="displayVerticalDrawingGridEvery" type="CT_DecimalNumber"
              minOccurs="0"/>
            <xsd:element name="doNotUseMarginsForDrawingGridOrigin" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="drawingGridHorizontalOrigin" type="CT_TwipsMeasure" minOccurs="0"/>
            <xsd:element name="drawingGridVerticalOrigin" type="CT_TwipsMeasure" minOccurs="0"/>
            <xsd:element name="doNotShadeFormData" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="noPunctuationKerning" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="characterSpacingControl" type="CT_CharacterSpacing" minOccurs="0"/>
            <xsd:element name="printTwoOnOne" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="strictFirstAndLastChars" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="noLineBreaksAfter" type="CT_Kinsoku" minOccurs="0"/>
            <xsd:element name="noLineBreaksBefore" type="CT_Kinsoku" minOccurs="0"/>
            <xsd:element name="savePreviewPicture" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotValidateAgainstSchema" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="saveInvalidXml" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="ignoreMixedContent" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="alwaysShowPlaceholderText" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotDemarcateInvalidXml" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="saveXmlDataOnly" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="useXSLTWhenSaving" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="saveThroughXslt" type="CT_SaveThroughXslt" minOccurs="0"/>
            <xsd:element name="showXMLTags" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="alwaysMergeEmptyNamespace" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="updateFields" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="hdrShapeDefaults" type="CT_ShapeDefaults" minOccurs="0"/>
            <xsd:element name="footnotePr" type="CT_FtnDocProps" minOccurs="0"/>
            <xsd:element name="endnotePr" type="CT_EdnDocProps" minOccurs="0"/>
            <xsd:element name="compat" type="CT_Compat" minOccurs="0"/>
            <xsd:element name="docVars" type="CT_DocVars" minOccurs="0"/>
            <xsd:element name="rsids" type="CT_DocRsids" minOccurs="0"/>
            <xsd:element ref="m:mathPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="attachedSchema" type="CT_String" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="themeFontLang" type="CT_Language" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="clrSchemeMapping" type="CT_ColorSchemeMapping" minOccurs="0"/>
            <xsd:element name="doNotIncludeSubdocsInStats" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotAutoCompressPictures" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="forceUpgrade" type="CT_Empty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="captions" type="CT_Captions" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="readModeInkLockDown" type="CT_ReadingModeInkLockDown" minOccurs="0"/>
            <xsd:element name="smartTagType" type="CT_SmartTagType" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element ref="sl:schemaLibrary" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="shapeDefaults" type="CT_ShapeDefaults" minOccurs="0"/>
            <xsd:element name="doNotEmbedSmartTags" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="decimalSymbol" type="CT_String" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="listSeparator" type="CT_String" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
```

```
    <xsd:complexType name="CT_StyleSort">
        <xsd:attribute name="val" type="ST_StyleSort" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_StylePaneFilter">
        <xsd:attribute name="allStyles" type="s:ST_OnOff"/>
        <xsd:attribute name="customStyles" type="s:ST_OnOff"/>
        <xsd:attribute name="latentStyles" type="s:ST_OnOff"/>
        <xsd:attribute name="stylesInUse" type="s:ST_OnOff"/>
        <xsd:attribute name="headingStyles" type="s:ST_OnOff"/>
        <xsd:attribute name="numberingStyles" type="s:ST_OnOff"/>
        <xsd:attribute name="tableStyles" type="s:ST_OnOff"/>
        <xsd:attribute name="directFormattingOnRuns" type="s:ST_OnOff"/>
        <xsd:attribute name="directFormattingOnParagraphs" type="s:ST_OnOff"/>
        <xsd:attribute name="directFormattingOnNumbering" type="s:ST_OnOff"/>
        <xsd:attribute name="directFormattingOnTables" type="s:ST_OnOff"/>
        <xsd:attribute name="clearFormatting" type="s:ST_OnOff"/>
        <xsd:attribute name="top3HeadingStyles" type="s:ST_OnOff"/>
        <xsd:attribute name="visibleStyles" type="s:ST_OnOff"/>
        <xsd:attribute name="alternateStyleNames" type="s:ST_OnOff"/>
        <xsd:attribute name="val" type="ST_ShortHexNumber"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_StyleSort">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="name"/>
            <xsd:enumeration value="priority"/>
            <xsd:enumeration value="default"/>
            <xsd:enumeration value="font"/>
            <xsd:enumeration value="basedOn"/>
            <xsd:enumeration value="type"/>
            <xsd:enumeration value="0000"/>
            <xsd:enumeration value="0001"/>
            <xsd:enumeration value="0002"/>
            <xsd:enumeration value="0003"/>
            <xsd:enumeration value="0004"/>
            <xsd:enumeration value="0005"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_WebSettings">
        <xsd:sequence>
            <xsd:element name="frameset" type="CT_Frameset" minOccurs="0"/>
            <xsd:element name="divs" type="CT_Divs" minOccurs="0"/>
            <xsd:element name="encoding" type="CT_String" minOccurs="0"/>
            <xsd:element name="optimizeForBrowser" type="CT_OptimizeForBrowser" minOccurs="0"/>
            <xsd:element name="relyOnVML" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="allowPNG" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotRelyOnCSS" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotSaveAsSingleFile" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotOrganizeInFolder" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotUseLongFileNames" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="pixelsPerInch" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="targetScreenSz" type="CT_TargetScreenSz" minOccurs="0"/>
            <xsd:element name="saveSmartTagsAsXml" type="CT_OnOff" minOccurs="0"/>
        </xsd:sequence>
```



```
    </xsd:complexType>
    <xsd:simpleType name="ST_FrameScrollbar">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="on"/>
            <xsd:enumeration value="off"/>
            <xsd:enumeration value="auto"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_FrameScrollbar">
        <xsd:attribute name="val" type="ST_FrameScrollbar" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OptimizeForBrowser">
        <xsd:complexContent>
            <xsd:extension base="CT_OnOff">
                <xsd:attribute name="target" type="s:ST_String" use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_Frame">
        <xsd:sequence>
            <xsd:element name="sz" type="CT_String" minOccurs="0"/>
            <xsd:element name="name" type="CT_String" minOccurs="0"/>
            <xsd:element name="title" type="CT_String" minOccurs="0"/>
            <xsd:element name="longDesc" type="CT_Rel" minOccurs="0"/>
            <xsd:element name="sourceFileName" type="CT_Rel" minOccurs="0"/>
            <xsd:element name="marW" type="CT_PixelsMeasure" minOccurs="0"/>
            <xsd:element name="marH" type="CT_PixelsMeasure" minOccurs="0"/>
            <xsd:element name="scrollbar" type="CT_FrameScrollbar" minOccurs="0"/>
            <xsd:element name="noResizeAllowed" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="linkedToFile" type="CT_OnOff" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_FrameLayout">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="rows"/>
            <xsd:enumeration value="cols"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_FrameLayout">
        <xsd:attribute name="val" type="ST_FrameLayout" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FramesetSplitbar">
        <xsd:sequence>
            <xsd:element name="w" type="CT_TwipsMeasure" minOccurs="0"/>
            <xsd:element name="color" type="CT_Color" minOccurs="0"/>
            <xsd:element name="noBorder" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="flatBorders" type="CT_OnOff" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Frameset">
        <xsd:sequence>
            <xsd:element name="sz" type="CT_String" minOccurs="0"/>
```

```
            <xsd:element name="framesetSplitbar" type="CT_FramesetSplitbar" minOccurs="0"/>
            <xsd:element name="frameLayout" type="CT_FrameLayout" minOccurs="0"/>
            <xsd:element name="title" type="CT_String" minOccurs="0"/>
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="frameset" type="CT_Frameset" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element name="frame" type="CT_Frame" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_NumPicBullet">
        <xsd:choice>
            <xsd:element name="pict" type="CT_Picture"/>
            <xsd:element name="drawing" type="CT_Drawing"/>
        </xsd:choice>
        <xsd:attribute name="numPicBulletId" type="ST_DecimalNumber" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_LevelSuffix">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="tab"/>
            <xsd:enumeration value="space"/>
            <xsd:enumeration value="nothing"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_LevelSuffix">
        <xsd:attribute name="val" type="ST_LevelSuffix" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_LevelText">
        <xsd:attribute name="val" type="s:ST_String" use="optional"/>
        <xsd:attribute name="null" type="s:ST_OnOff" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_LvlLegacy">
        <xsd:attribute name="legacy" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="legacySpace" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="legacyIndent" type="ST_SignedTwipsMeasure" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Lvl">
        <xsd:sequence>
            <xsd:element name="start" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="numFmt" type="CT_NumFmt" minOccurs="0"/>
            <xsd:element name="lvlRestart" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="pStyle" type="CT_String" minOccurs="0"/>
            <xsd:element name="isLgl" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="suff" type="CT_LevelSuffix" minOccurs="0"/>
            <xsd:element name="lvlText" type="CT_LevelText" minOccurs="0"/>
            <xsd:element name="lvlPicBulletId" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="legacy" type="CT_LvlLegacy" minOccurs="0"/>
            <xsd:element name="lvlJc" type="CT_Jc" minOccurs="0"/>
            <xsd:element name="pPr" type="CT_PPrGeneral" minOccurs="0"/>
            <xsd:element name="rPr" type="CT_RPr" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="ilvl" type="ST_DecimalNumber" use="required"/>
        <xsd:attribute name="tplc" type="ST_LongHexNumber" use="optional"/>
        <xsd:attribute name="tentative" type="s:ST_OnOff" use="optional"/>
```



```
    </xsd:complexType>
    <xsd:simpleType name="ST_MultiLevelType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="singleLevel"/>
            <xsd:enumeration value="multilevel"/>
            <xsd:enumeration value="hybridMultilevel"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_MultiLevelType">
        <xsd:attribute name="val" type="ST_MultiLevelType" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AbstractNum">
        <xsd:sequence>
            <xsd:element name="nsid" type="CT_LongHexNumber" minOccurs="0"/>
            <xsd:element name="multiLevelType" type="CT_MultiLevelType" minOccurs="0"/>
            <xsd:element name="tmpl" type="CT_LongHexNumber" minOccurs="0"/>
            <xsd:element name="name" type="CT_String" minOccurs="0"/>
            <xsd:element name="styleLink" type="CT_String" minOccurs="0"/>
            <xsd:element name="numStyleLink" type="CT_String" minOccurs="0"/>
            <xsd:element name="lvl" type="CT_Lvl" minOccurs="0" maxOccurs="9"/>
        </xsd:sequence>
        <xsd:attribute name="abstractNumId" type="ST_DecimalNumber" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NumLvl">
        <xsd:sequence>
            <xsd:element name="startOverride" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="lvl" type="CT_Lvl" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="ilvl" type="ST_DecimalNumber" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Num">
        <xsd:sequence>
            <xsd:element name="abstractNumId" type="CT_DecimalNumber" minOccurs="1"/>
            <xsd:element name="lvlOverride" type="CT_NumLvl" minOccurs="0" maxOccurs="9"/>
        </xsd:sequence>
        <xsd:attribute name="numId" type="ST_DecimalNumber" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Numbering">
        <xsd:sequence>
            <xsd:element name="numPicBullet" type="CT_NumPicBullet" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="abstractNum" type="CT_AbstractNum" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="num" type="CT_Num" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="numIdMacAtCleanup" type="CT_DecimalNumber" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_TblStyleOverrideType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="wholeTable"/>
            <xsd:enumeration value="firstRow"/>
            <xsd:enumeration value="lastRow"/>
            <xsd:enumeration value="firstCol"/>
```

```
            <xsd:enumeration value="lastCol"/>
            <xsd:enumeration value="band1Vert"/>
            <xsd:enumeration value="band2Vert"/>
            <xsd:enumeration value="band1Horz"/>
            <xsd:enumeration value="band2Horz"/>
            <xsd:enumeration value="neCell"/>
            <xsd:enumeration value="nwCell"/>
            <xsd:enumeration value="seCell"/>
            <xsd:enumeration value="swCell"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TblStylePr">
        <xsd:sequence>
            <xsd:element name="pPr" type="CT_PPrGeneral" minOccurs="0"/>
            <xsd:element name="rPr" type="CT_RPr" minOccurs="0"/>
            <xsd:element name="tblPr" type="CT_TblPrBase" minOccurs="0"/>
            <xsd:element name="trPr" type="CT_TrPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tcPr" type="CT_TcPr" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="type" type="ST_TblStyleOverrideType" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_StyleType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="paragraph"/>
            <xsd:enumeration value="character"/>
            <xsd:enumeration value="table"/>
            <xsd:enumeration value="numbering"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Style">
        <xsd:sequence>
            <xsd:element name="name" type="CT_String" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="aliases" type="CT_String" minOccurs="0"/>
            <xsd:element name="basedOn" type="CT_String" minOccurs="0"/>
            <xsd:element name="next" type="CT_String" minOccurs="0"/>
            <xsd:element name="link" type="CT_String" minOccurs="0"/>
            <xsd:element name="autoRedefine" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="hidden" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="uiPriority" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="semiHidden" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="unhideWhenUsed" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="qFormat" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="locked" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="personal" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="personalCompose" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="personalReply" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="rsid" type="CT_LongHexNumber" minOccurs="0"/>
            <xsd:element name="pPr" type="CT_PPrGeneral" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="rPr" type="CT_RPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblPr" type="CT_TblPrBase" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="trPr" type="CT_TrPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tcPr" type="CT_TcPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblStylePr" type="CT_TblStylePr" minOccurs="0" maxOccurs="unbounded"/>
```



```
        </xsd:sequence>
        <xsd:attribute name="type" type="ST_StyleType" use="optional"/>
        <xsd:attribute name="styleId" type="s:ST_String" use="optional"/>
        <xsd:attribute name="default" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="customStyle" type="s:ST_OnOff" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_LsdException">
        <xsd:attribute name="name" type="s:ST_String" use="required"/>
        <xsd:attribute name="locked" type="s:ST_OnOff"/>
        <xsd:attribute name="uiPriority" type="ST_DecimalNumber"/>
        <xsd:attribute name="semiHidden" type="s:ST_OnOff"/>
        <xsd:attribute name="unhideWhenUsed" type="s:ST_OnOff"/>
        <xsd:attribute name="qFormat" type="s:ST_OnOff"/>
    </xsd:complexType>
    <xsd:complexType name="CT_LatentStyles">
        <xsd:sequence>
            <xsd:element name="lsdException" type="CT_LsdException" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="defLockedState" type="s:ST_OnOff"/>
        <xsd:attribute name="defUIPriority" type="ST_DecimalNumber"/>
        <xsd:attribute name="defSemiHidden" type="s:ST_OnOff"/>
        <xsd:attribute name="defUnhideWhenUsed" type="s:ST_OnOff"/>
        <xsd:attribute name="defQFormat" type="s:ST_OnOff"/>
        <xsd:attribute name="count" type="ST_DecimalNumber"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Styles">
        <xsd:sequence>
            <xsd:element name="docDefaults" type="CT_DocDefaults" minOccurs="0"/>
            <xsd:element name="latentStyles" type="CT_LatentStyles" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="style" type="CT_Style" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Panose">
        <xsd:attribute name="val" type="s:ST_Panose" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_FontFamily">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="decorative"/>
            <xsd:enumeration value="modern"/>
            <xsd:enumeration value="roman"/>
            <xsd:enumeration value="script"/>
            <xsd:enumeration value="swiss"/>
            <xsd:enumeration value="auto"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_FontFamily">
        <xsd:attribute name="val" type="ST_FontFamily" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Pitch">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="fixed"/>
            <xsd:enumeration value="variable"/>
```

```
            <xsd:enumeration value="default"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Pitch">
        <xsd:attribute name="val" type="ST_Pitch" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FontSig">
        <xsd:attribute name="usb0" use="required" type="ST_LongHexNumber"/>
        <xsd:attribute name="usb1" use="required" type="ST_LongHexNumber"/>
        <xsd:attribute name="usb2" use="required" type="ST_LongHexNumber"/>
        <xsd:attribute name="usb3" use="required" type="ST_LongHexNumber"/>
        <xsd:attribute name="csb0" use="required" type="ST_LongHexNumber"/>
        <xsd:attribute name="csb1" use="required" type="ST_LongHexNumber"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FontRel">
        <xsd:complexContent>
            <xsd:extension base="CT_Rel">
                <xsd:attribute name="fontKey" type="s:ST_Guid"/>
                <xsd:attribute name="subsetted" type="s:ST_OnOff"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_Font">
        <xsd:sequence>
            <xsd:element name="altName" type="CT_String" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="panose1" type="CT_Panose" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="charset" type="CT_Charset" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="family" type="CT_FontFamily" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="notTrueType" type="CT_OnOff" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pitch" type="CT_Pitch" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sig" type="CT_FontSig" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="embedRegular" type="CT_FontRel" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="embedBold" type="CT_FontRel" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="embedItalic" type="CT_FontRel" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="embedBoldItalic" type="CT_FontRel" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="s:ST_String" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FontsList">
        <xsd:sequence>
            <xsd:element name="font" type="CT_Font" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DivBdr">
        <xsd:sequence>
            <xsd:element name="top" type="CT_Border" minOccurs="0"/>
            <xsd:element name="left" type="CT_Border" minOccurs="0"/>
            <xsd:element name="bottom" type="CT_Border" minOccurs="0"/>
            <xsd:element name="right" type="CT_Border" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Div">
        <xsd:sequence>
```



```
            <xsd:element name="blockQuote" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="bodyDiv" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="marLeft" type="CT_SignedTwipsMeasure"/>
            <xsd:element name="marRight" type="CT_SignedTwipsMeasure"/>
            <xsd:element name="marTop" type="CT_SignedTwipsMeasure"/>
            <xsd:element name="marBottom" type="CT_SignedTwipsMeasure"/>
            <xsd:element name="divBdr" type="CT_DivBdr" minOccurs="0"/>
            <xsd:element name="divsChild" type="CT_Divs" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="ST_DecimalNumber" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Divs">
        <xsd:sequence minOccurs="1" maxOccurs="unbounded">
            <xsd:element name="div" type="CT_Div"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TxbxContent">
        <xsd:group ref="EG_BlockLevelElts" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:complexType>
    <xsd:element name="txbxContent" type="CT_TxbxContent"/>
    <xsd:group name="EG_MathContent">
        <xsd:choice>
            <xsd:element ref="m:oMathPara"/>
            <xsd:element ref="m:oMath"/>
        </xsd:choice>
    </xsd:group>
    <xsd:group name="EG_BlockLevelChunkElts">
        <xsd:choice>
            <xsd:group ref="EG_ContentBlockContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:group>
    <xsd:group name="EG_BlockLevelElts">
        <xsd:choice>
            <xsd:group ref="EG_BlockLevelChunkElts" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="altChunk" type="CT_AltChunk" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:group>
    <xsd:group name="EG_RunLevelElts">
        <xsd:choice>
            <xsd:element name="proofErr" minOccurs="0" type="CT_ProofErr"/>
            <xsd:element name="permStart" minOccurs="0" type="CT_PermStart"/>
            <xsd:element name="permEnd" minOccurs="0" type="CT_Perm"/>
            <xsd:group ref="EG_RangeMarkupElements" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="ins" type="CT_RunTrackChange" minOccurs="0"/>
            <xsd:element name="del" type="CT_RunTrackChange" minOccurs="0"/>
            <xsd:element name="moveFrom" type="CT_RunTrackChange"/>
            <xsd:element name="moveTo" type="CT_RunTrackChange"/>
            <xsd:group ref="EG_MathContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_Body">
        <xsd:sequence>
            <xsd:group ref="EG_BlockLevelElts" minOccurs="0" maxOccurs="unbounded"/>
```

```
            <xsd:element name="sectPr" minOccurs="0" maxOccurs="1" type="CT_SectPr"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ShapeDefaults">
        <xsd:choice maxOccurs="unbounded">
            <xsd:any processContents="lax" namespace="urn:schemas-microsoft-com:office:office"
              minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_Comments">
        <xsd:sequence>
            <xsd:element name="comment" type="CT_Comment" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="comments" type="CT_Comments"/>
    <xsd:complexType name="CT_Footnotes">
        <xsd:sequence maxOccurs="unbounded">
            <xsd:element name="footnote" type="CT_FtnEdn" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="footnotes" type="CT_Footnotes"/>
    <xsd:complexType name="CT_Endnotes">
        <xsd:sequence maxOccurs="unbounded">
            <xsd:element name="endnote" type="CT_FtnEdn" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="endnotes" type="CT_Endnotes"/>
    <xsd:element name="hdr" type="CT_HdrFtr"/>
    <xsd:element name="ftr" type="CT_HdrFtr"/>
    <xsd:complexType name="CT_SmartTagType">
        <xsd:attribute name="namespaceuri" type="s:ST_String"/>
        <xsd:attribute name="name" type="s:ST_String"/>
        <xsd:attribute name="url" type="s:ST_String"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_ThemeColor">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="dark1"/>
            <xsd:enumeration value="light1"/>
            <xsd:enumeration value="dark2"/>
            <xsd:enumeration value="light2"/>
            <xsd:enumeration value="accent1"/>
            <xsd:enumeration value="accent2"/>
            <xsd:enumeration value="accent3"/>
            <xsd:enumeration value="accent4"/>
            <xsd:enumeration value="accent5"/>
            <xsd:enumeration value="accent6"/>
            <xsd:enumeration value="hyperlink"/>
            <xsd:enumeration value="followedHyperlink"/>
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="background1"/>
            <xsd:enumeration value="text1"/>
            <xsd:enumeration value="background2"/>
            <xsd:enumeration value="text2"/>
```



```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_DocPartBehavior">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="content"/>
            <xsd:enumeration value="p"/>
            <xsd:enumeration value="pg"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DocPartBehavior">
        <xsd:attribute name="val" use="required" type="ST_DocPartBehavior"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DocPartBehaviors">
        <xsd:choice>
            <xsd:element name="behavior" type="CT_DocPartBehavior" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:simpleType name="ST_DocPartType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="normal"/>
            <xsd:enumeration value="autoExp"/>
            <xsd:enumeration value="toolbar"/>
            <xsd:enumeration value="speller"/>
            <xsd:enumeration value="formFld"/>
            <xsd:enumeration value="bbPlcHdr"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DocPartType">
        <xsd:attribute name="val" use="required" type="ST_DocPartType"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DocPartTypes">
        <xsd:choice>
            <xsd:element name="type" type="CT_DocPartType" maxOccurs="unbounded"/>
        </xsd:choice>
        <xsd:attribute name="all" type="s:ST_OnOff" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DocPartGallery">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="placeholder"/>
            <xsd:enumeration value="any"/>
            <xsd:enumeration value="default"/>
            <xsd:enumeration value="docParts"/>
            <xsd:enumeration value="coverPg"/>
            <xsd:enumeration value="eq"/>
            <xsd:enumeration value="ftrs"/>
            <xsd:enumeration value="hdrs"/>
            <xsd:enumeration value="pgNum"/>
            <xsd:enumeration value="tbls"/>
            <xsd:enumeration value="watermarks"/>
            <xsd:enumeration value="autoTxt"/>
            <xsd:enumeration value="txtBox"/>
            <xsd:enumeration value="pgNumT"/>
```

```
            <xsd:enumeration value="pgNumB"/>
            <xsd:enumeration value="pgNumMargins"/>
            <xsd:enumeration value="tblOfContents"/>
            <xsd:enumeration value="bib"/>
            <xsd:enumeration value="custQuickParts"/>
            <xsd:enumeration value="custCoverPg"/>
            <xsd:enumeration value="custEq"/>
            <xsd:enumeration value="custFtrs"/>
            <xsd:enumeration value="custHdrs"/>
            <xsd:enumeration value="custPgNum"/>
            <xsd:enumeration value="custTbls"/>
            <xsd:enumeration value="custWatermarks"/>
            <xsd:enumeration value="custAutoTxt"/>
            <xsd:enumeration value="custTxtBox"/>
            <xsd:enumeration value="custPgNumT"/>
            <xsd:enumeration value="custPgNumB"/>
            <xsd:enumeration value="custPgNumMargins"/>
            <xsd:enumeration value="custTblOfContents"/>
            <xsd:enumeration value="custBib"/>
            <xsd:enumeration value="custom1"/>
            <xsd:enumeration value="custom2"/>
            <xsd:enumeration value="custom3"/>
            <xsd:enumeration value="custom4"/>
            <xsd:enumeration value="custom5"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DocPartGallery">
        <xsd:attribute name="val" type="ST_DocPartGallery" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DocPartCategory">
        <xsd:sequence>
            <xsd:element name="name" type="CT_String" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="gallery" type="CT_DocPartGallery" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DocPartName">
        <xsd:attribute name="val" type="s:ST_String" use="required"/>
        <xsd:attribute name="decorated" type="s:ST_OnOff" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DocPartPr">
        <xsd:all>
            <xsd:element name="style" type="CT_String" minOccurs="0"/>
            <xsd:element name="category" type="CT_DocPartCategory" minOccurs="0"/>
            <xsd:element name="types" type="CT_DocPartTypes" minOccurs="0"/>
            <xsd:element name="behaviors" type="CT_DocPartBehaviors" minOccurs="0"/>
            <xsd:element name="description" type="CT_String" minOccurs="0"/>
            <xsd:element name="guid" type="CT_Guid" minOccurs="0"/>
        </xsd:all>
    </xsd:complexType>
    <xsd:complexType name="CT_DocPart">
        <xsd:sequence>
            <xsd:element name="docPartPr" type="CT_DocPartPr" minOccurs="0"/>
            <xsd:element name="docPartBody" type="CT_Body" minOccurs="0"/>
```



```
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DocParts">
        <xsd:choice>
            <xsd:element name="docPart" type="CT_DocPart" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:element name="settings" type="CT_Settings"/>
    <xsd:element name="webSettings" type="CT_WebSettings"/>
    <xsd:element name="fonts" type="CT_FontsList"/>
    <xsd:element name="numbering" type="CT_Numbering"/>
    <xsd:element name="styles" type="CT_Styles"/>
    <xsd:simpleType name="ST_CaptionPos">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="above"/>
            <xsd:enumeration value="below"/>
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="right"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Caption">
        <xsd:attribute name="name" type="s:ST_String" use="required"/>
        <xsd:attribute name="pos" type="ST_CaptionPos" use="optional"/>
        <xsd:attribute name="chapNum" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="heading" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="noLabel" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="numFmt" type="ST_NumberFormat" use="optional"/>
        <xsd:attribute name="sep" type="ST_ChapterSep" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AutoCaption">
        <xsd:attribute name="name" type="s:ST_String" use="required"/>
        <xsd:attribute name="caption" type="s:ST_String" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AutoCaptions">
        <xsd:sequence>
            <xsd:element name="autoCaption" type="CT_AutoCaption" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Captions">
        <xsd:sequence>
            <xsd:element name="caption" type="CT_Caption" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="autoCaptions" type="CT_AutoCaptions" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DocumentBase">
        <xsd:sequence>
            <xsd:element name="background" type="CT_Background" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Document">
        <xsd:complexContent>
            <xsd:extension base="CT_DocumentBase">
```

```
                <xsd:sequence>
                    <xsd:element name="body" type="CT_Body" minOccurs="0" maxOccurs="1"/>
                </xsd:sequence>
                <xsd:attribute name="conformance" type="s:ST_ConformanceClass"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_GlossaryDocument">
        <xsd:complexContent>
            <xsd:extension base="CT_DocumentBase">
                <xsd:sequence>
                    <xsd:element name="docParts" type="CT_DocParts" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="document" type="CT_Document"/>
    <xsd:element name="glossaryDocument" type="CT_GlossaryDocument"/>
</xsd:schema>
```

A.2 SpreadsheetML

This schema is available in the file sml.xsd.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main"
  xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
  xmlns:xdr="http://schemas.openxmlformats.org/drawingml/2006/spreadsheetDrawing"
  xmlns:s="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
  targetNamespace="http://schemas.openxmlformats.org/spreadsheetml/2006/main"
  elementFormDefault="qualified">
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
      schemaLocation="shared-relationshipReference.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/drawingml/2006/spreadsheetDrawing"
      schemaLocation="dml-spreadsheetDrawing.xsd"/>
    <xsd:complexType name="CT_AutoFilter">
        <xsd:sequence>
            <xsd:element name="filterColumn" minOccurs="0" maxOccurs="unbounded"
              type="CT_FilterColumn"/>
            <xsd:element name="sortState" minOccurs="0" maxOccurs="1" type="CT_SortState"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="ref" type="ST_Ref"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FilterColumn">
        <xsd:choice minOccurs="0" maxOccurs="1">
            <xsd:element name="filters" type="CT_Filters" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="top10" type="CT_Top10" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="customFilters" type="CT_CustomFilters" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dynamicFilter" type="CT_DynamicFilter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="colorFilter" type="CT_ColorFilter" minOccurs="0" maxOccurs="1"/>
```



```
            <xsd:element name="iconFilter" minOccurs="0" maxOccurs="1" type="CT_IconFilter"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:choice>
        <xsd:attribute name="colId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="hiddenButton" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showButton" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Filters">
        <xsd:sequence>
            <xsd:element name="filter" type="CT_Filter" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="dateGroupItem" type="CT_DateGroupItem" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="blank" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="calendarType" type="s:ST_CalendarType" use="optional" default="none"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Filter">
        <xsd:attribute name="val" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomFilters">
        <xsd:sequence>
            <xsd:element name="customFilter" type="CT_CustomFilter" minOccurs="1" maxOccurs="2"/>
        </xsd:sequence>
        <xsd:attribute name="and" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomFilter">
        <xsd:attribute name="operator" type="ST_FilterOperator" default="equal" use="optional"/>
        <xsd:attribute name="val" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Top10">
        <xsd:attribute name="top" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="percent" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="val" type="xsd:double" use="required"/>
        <xsd:attribute name="filterVal" type="xsd:double" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ColorFilter">
        <xsd:attribute name="dxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="cellColor" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_IconFilter">
        <xsd:attribute name="iconSet" type="ST_IconSetType" use="required"/>
        <xsd:attribute name="iconId" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_FilterOperator">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="equal"/>
            <xsd:enumeration value="lessThan"/>
            <xsd:enumeration value="lessThanOrEqual"/>
            <xsd:enumeration value="notEqual"/>
            <xsd:enumeration value="greaterThanOrEqual"/>
            <xsd:enumeration value="greaterThan"/>
        </xsd:restriction>
    </xsd:simpleType>
```

```
    <xsd:complexType name="CT_DynamicFilter">
        <xsd:attribute name="type" type="ST_DynamicFilterType" use="required"/>
        <xsd:attribute name="val" type="xsd:double" use="optional"/>
        <xsd:attribute name="valIso" type="xsd:dateTime" use="optional"/>
        <xsd:attribute name="maxVal" type="xsd:double" use="optional"/>
        <xsd:attribute name="maxValIso" type="xsd:dateTime" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DynamicFilterType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="null"/>
            <xsd:enumeration value="aboveAverage"/>
            <xsd:enumeration value="belowAverage"/>
            <xsd:enumeration value="tomorrow"/>
            <xsd:enumeration value="today"/>
            <xsd:enumeration value="yesterday"/>
            <xsd:enumeration value="nextWeek"/>
            <xsd:enumeration value="thisWeek"/>
            <xsd:enumeration value="lastWeek"/>
            <xsd:enumeration value="nextMonth"/>
            <xsd:enumeration value="thisMonth"/>
            <xsd:enumeration value="lastMonth"/>
            <xsd:enumeration value="nextQuarter"/>
            <xsd:enumeration value="thisQuarter"/>
            <xsd:enumeration value="lastQuarter"/>
            <xsd:enumeration value="nextYear"/>
            <xsd:enumeration value="thisYear"/>
            <xsd:enumeration value="lastYear"/>
            <xsd:enumeration value="yearToDate"/>
            <xsd:enumeration value="Q1"/>
            <xsd:enumeration value="Q2"/>
            <xsd:enumeration value="Q3"/>
            <xsd:enumeration value="Q4"/>
            <xsd:enumeration value="M1"/>
            <xsd:enumeration value="M2"/>
            <xsd:enumeration value="M3"/>
            <xsd:enumeration value="M4"/>
            <xsd:enumeration value="M5"/>
            <xsd:enumeration value="M6"/>
            <xsd:enumeration value="M7"/>
            <xsd:enumeration value="M8"/>
            <xsd:enumeration value="M9"/>
            <xsd:enumeration value="M10"/>
            <xsd:enumeration value="M11"/>
            <xsd:enumeration value="M12"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_IconSetType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="3Arrows"/>
            <xsd:enumeration value="3ArrowsGray"/>
            <xsd:enumeration value="3Flags"/>
            <xsd:enumeration value="3TrafficLights1"/>
            <xsd:enumeration value="3TrafficLights2"/>
```



```
            <xsd:enumeration value="3Signs"/>
            <xsd:enumeration value="3Symbols"/>
            <xsd:enumeration value="3Symbols2"/>
            <xsd:enumeration value="4Arrows"/>
            <xsd:enumeration value="4ArrowsGray"/>
            <xsd:enumeration value="4RedToBlack"/>
            <xsd:enumeration value="4Rating"/>
            <xsd:enumeration value="4TrafficLights"/>
            <xsd:enumeration value="5Arrows"/>
            <xsd:enumeration value="5ArrowsGray"/>
            <xsd:enumeration value="5Rating"/>
            <xsd:enumeration value="5Quarters"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SortState">
        <xsd:sequence>
            <xsd:element name="sortCondition" minOccurs="0" maxOccurs="64" type="CT_SortCondition"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="columnSort" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="caseSensitive" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="sortMethod" type="ST_SortMethod" use="optional" default="none"/>
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SortCondition">
        <xsd:attribute name="descending" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="sortBy" type="ST_SortBy" use="optional" default="value"/>
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
        <xsd:attribute name="customList" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="dxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="iconSet" type="ST_IconSetType" use="optional" default="3Arrows"/>
        <xsd:attribute name="iconId" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SortBy">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="value"/>
            <xsd:enumeration value="cellColor"/>
            <xsd:enumeration value="fontColor"/>
            <xsd:enumeration value="icon"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_SortMethod">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="stroke"/>
            <xsd:enumeration value="pinYin"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DateGroupItem">
        <xsd:attribute name="year" type="xsd:unsignedShort" use="required"/>
        <xsd:attribute name="month" type="xsd:unsignedShort" use="optional"/>
        <xsd:attribute name="day" type="xsd:unsignedShort" use="optional"/>
        <xsd:attribute name="hour" type="xsd:unsignedShort" use="optional"/>
```

```
        <xsd:attribute name="minute" type="xsd:unsignedShort" use="optional"/>
        <xsd:attribute name="second" type="xsd:unsignedShort" use="optional"/>
        <xsd:attribute name="dateTimeGrouping" type="ST_DateTimeGrouping" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DateTimeGrouping">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="year"/>
            <xsd:enumeration value="month"/>
            <xsd:enumeration value="day"/>
            <xsd:enumeration value="hour"/>
            <xsd:enumeration value="minute"/>
            <xsd:enumeration value="second"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_CellRef">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Ref">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_RefA">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Sqref">
        <xsd:list itemType="ST_Ref"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Formula">
        <xsd:restriction base="s:ST_Xstring"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_UnsignedIntHex">
        <xsd:restriction base="xsd:hexBinary">
            <xsd:length value="4"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_UnsignedShortHex">
        <xsd:restriction base="xsd:hexBinary">
            <xsd:length value="2"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_XStringElement">
        <xsd:attribute name="v" type="s:ST_Xstring" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Extension">
        <xsd:sequence>
            <xsd:any processContents="lax"/>
        </xsd:sequence>
        <xsd:attribute name="uri" type="xsd:token"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ObjectAnchor">
        <xsd:sequence>
            <xsd:element ref="xdr:from" minOccurs="1" maxOccurs="1"/>
            <xsd:element ref="xdr:to" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
```



```
        <xsd:attribute name="moveWithCells" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="sizeWithCells" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:group name="EG_ExtensionList">
        <xsd:sequence>
            <xsd:element name="ext" type="CT_Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_ExtensionList">
        <xsd:sequence>
            <xsd:group ref="EG_ExtensionList" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="calcChain" type="CT_CalcChain"/>
    <xsd:complexType name="CT_CalcChain">
        <xsd:sequence>
            <xsd:element name="c" type="CT_CalcCell" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CalcCell">
        <xsd:attribute name="r" type="ST_CellRef" use="optional"/>
        <xsd:attribute name="ref" type="ST_CellRef" use="optional"/>
        <xsd:attribute name="i" type="xsd:int" use="optional" default="0"/>
        <xsd:attribute name="s" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="l" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="t" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="a" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:element name="comments" type="CT_Comments"/>
    <xsd:complexType name="CT_Comments">
        <xsd:sequence>
            <xsd:element name="authors" type="CT_Authors" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="commentList" type="CT_CommentList" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Authors">
        <xsd:sequence>
            <xsd:element name="author" type="s:ST_Xstring" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CommentList">
        <xsd:sequence>
            <xsd:element name="comment" type="CT_Comment" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Comment">
        <xsd:sequence>
            <xsd:element name="text" type="CT_Rst" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="commentPr" type="CT_CommentPr" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
```

```
        <xsd:attribute name="authorId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="guid" type="s:ST_Guid" use="optional"/>
        <xsd:attribute name="shapeId" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CommentPr">
        <xsd:sequence>
            <xsd:element name="anchor" type="CT_ObjectAnchor" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="locked" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="defaultSize" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="print" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="disabled" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="autoFill" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="autoLine" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="altText" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="textHAlign" type="ST_TextHAlign" use="optional" default="left"/>
        <xsd:attribute name="textVAlign" type="ST_TextVAlign" use="optional" default="top"/>
        <xsd:attribute name="lockText" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="justLastX" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="autoScale" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TextHAlign">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="right"/>
            <xsd:enumeration value="justify"/>
            <xsd:enumeration value="distributed"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextVAlign">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="top"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="bottom"/>
            <xsd:enumeration value="justify"/>
            <xsd:enumeration value="distributed"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="MapInfo" type="CT_MapInfo"/>
    <xsd:complexType name="CT_MapInfo">
        <xsd:sequence>
            <xsd:element name="Schema" type="CT_Schema" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="Map" type="CT_Map" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="SelectionNamespaces" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Schema" mixed="true">
        <xsd:sequence>
            <xsd:any/>
        </xsd:sequence>
        <xsd:attribute name="ID" type="xsd:string" use="required"/>
        <xsd:attribute name="SchemaRef" type="xsd:string" use="optional"/>
```



```
        <xsd:attribute name="Namespace" type="xsd:string" use="optional"/>
        <xsd:attribute name="SchemaLanguage" type="xsd:token" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Map">
        <xsd:sequence>
            <xsd:element name="DataBinding" type="CT_DataBinding" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="ID" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="Name" type="xsd:string" use="required"/>
        <xsd:attribute name="RootElement" type="xsd:string" use="required"/>
        <xsd:attribute name="SchemaID" type="xsd:string" use="required"/>
        <xsd:attribute name="ShowImportExportValidationErrors" type="xsd:boolean" use="required"/>
        <xsd:attribute name="AutoFit" type="xsd:boolean" use="required"/>
        <xsd:attribute name="Append" type="xsd:boolean" use="required"/>
        <xsd:attribute name="PreserveSortAFLayout" type="xsd:boolean" use="required"/>
        <xsd:attribute name="PreserveFormat" type="xsd:boolean" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DataBinding">
        <xsd:sequence>
            <xsd:any/>
        </xsd:sequence>
        <xsd:attribute name="DataBindingName" type="xsd:string" use="optional"/>
        <xsd:attribute name="FileBinding" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="ConnectionID" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="FileBindingName" type="xsd:string" use="optional"/>
        <xsd:attribute name="DataBindingLoadMode" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:element name="connections" type="CT_Connections"/>
    <xsd:complexType name="CT_Connections">
        <xsd:sequence>
            <xsd:element name="connection" minOccurs="1" maxOccurs="unbounded" type="CT_Connection"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Connection">
        <xsd:sequence>
            <xsd:element name="dbPr" minOccurs="0" maxOccurs="1" type="CT_DbPr"/>
            <xsd:element name="olapPr" minOccurs="0" maxOccurs="1" type="CT_OlapPr"/>
            <xsd:element name="webPr" minOccurs="0" maxOccurs="1" type="CT_WebPr"/>
            <xsd:element name="textPr" minOccurs="0" maxOccurs="1" type="CT_TextPr"/>
            <xsd:element name="parameters" minOccurs="0" maxOccurs="1" type="CT_Parameters"/>
            <xsd:element name="extLst" minOccurs="0" maxOccurs="1" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="id" use="required" type="xsd:unsignedInt"/>
        <xsd:attribute name="sourceFile" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="odcFile" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="keepAlive" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="interval" use="optional" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="name" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="description" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="type" use="optional" type="xsd:unsignedInt"/>
        <xsd:attribute name="reconnectionMethod" use="optional" type="xsd:unsignedInt" default="1"/>
        <xsd:attribute name="refreshedVersion" use="required" type="xsd:unsignedByte"/>
```

```
        <xsd:attribute name="minRefreshableVersion" use="optional" type="xsd:unsignedByte"
          default="0"/>
        <xsd:attribute name="savePassword" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="new" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="deleted" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="onlyUseConnectionFile" use="optional" type="xsd:boolean"
          default="false"/>
        <xsd:attribute name="background" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="refreshOnLoad" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="saveData" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="credentials" use="optional" type="ST_CredMethod" default="integrated"/>
        <xsd:attribute name="singleSignOnId" use="optional" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_CredMethod">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="integrated"/>
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="stored"/>
            <xsd:enumeration value="prompt"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DbPr">
        <xsd:attribute name="connection" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="command" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="serverCommand" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="commandType" use="optional" type="xsd:unsignedInt" default="2"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OlapPr">
        <xsd:attribute name="local" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="localConnection" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="localRefresh" use="optional" type="xsd:boolean" default="true"/>
        <xsd:attribute name="sendLocale" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="rowDrillCount" use="optional" type="xsd:unsignedInt"/>
        <xsd:attribute name="serverFill" use="optional" type="xsd:boolean" default="true"/>
        <xsd:attribute name="serverNumberFormat" use="optional" type="xsd:boolean" default="true"/>
        <xsd:attribute name="serverFont" use="optional" type="xsd:boolean" default="true"/>
        <xsd:attribute name="serverFontColor" use="optional" type="xsd:boolean" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WebPr">
        <xsd:sequence>
            <xsd:element name="tables" minOccurs="0" maxOccurs="1" type="CT_Tables"/>
        </xsd:sequence>
        <xsd:attribute name="xml" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="sourceData" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="parsePre" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="consecutive" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="firstRow" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="xl97" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="textDates" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="xl2000" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="url" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="post" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="htmlTables" use="optional" type="xsd:boolean" default="false"/>
```



```
        <xsd:attribute name="htmlFormat" use="optional" type="ST_HtmlFmt" default="none"/>
        <xsd:attribute name="editPage" use="optional" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_HtmlFmt">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="rtf"/>
            <xsd:enumeration value="all"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Parameters">
        <xsd:sequence>
            <xsd:element name="parameter" minOccurs="1" maxOccurs="unbounded" type="CT_Parameter"/>
        </xsd:sequence>
        <xsd:attribute name="count" use="optional" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Parameter">
        <xsd:attribute name="name" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="sqlType" use="optional" type="xsd:int" default="0"/>
        <xsd:attribute name="parameterType" use="optional" type="ST_ParameterType" default="prompt"/>
        <xsd:attribute name="refreshOnChange" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="prompt" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="boolean" use="optional" type="xsd:boolean"/>
        <xsd:attribute name="double" use="optional" type="xsd:double"/>
        <xsd:attribute name="integer" use="optional" type="xsd:int"/>
        <xsd:attribute name="string" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="cell" use="optional" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_ParameterType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="prompt"/>
            <xsd:enumeration value="value"/>
            <xsd:enumeration value="cell"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Tables">
        <xsd:choice minOccurs="1" maxOccurs="unbounded">
            <xsd:element name="m" type="CT_TableMissing"/>
            <xsd:element name="s" type="CT_XStringElement"/>
            <xsd:element name="x" type="CT_Index"/>
        </xsd:choice>
        <xsd:attribute name="count" use="optional" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TableMissing"/>
    <xsd:complexType name="CT_TextPr">
        <xsd:sequence>
            <xsd:element name="textFields" minOccurs="0" maxOccurs="1" type="CT_TextFields"/>
        </xsd:sequence>
        <xsd:attribute name="prompt" use="optional" type="xsd:boolean" default="true"/>
        <xsd:attribute name="fileType" use="optional" type="ST_FileType" default="win"/>
        <xsd:attribute name="codePage" use="optional" type="xsd:unsignedInt" default="1252"/>
        <xsd:attribute name="characterSet" use="optional" type="xsd:string"/>
        <xsd:attribute name="firstRow" use="optional" type="xsd:unsignedInt" default="1"/>
```

```
        <xsd:attribute name="sourceFile" use="optional" type="s:ST_Xstring" default=""/>
        <xsd:attribute name="delimited" use="optional" type="xsd:boolean" default="true"/>
        <xsd:attribute name="decimal" use="optional" type="s:ST_Xstring" default="."/>
        <xsd:attribute name="thousands" use="optional" type="s:ST_Xstring" default=","/>
        <xsd:attribute name="tab" use="optional" type="xsd:boolean" default="true"/>
        <xsd:attribute name="space" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="comma" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="semicolon" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="consecutive" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="qualifier" use="optional" type="ST_Qualifier" default="doubleQuote"/>
        <xsd:attribute name="delimiter" use="optional" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_FileType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="mac"/>
            <xsd:enumeration value="win"/>
            <xsd:enumeration value="dos"/>
            <xsd:enumeration value="lin"/>
            <xsd:enumeration value="other"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Qualifier">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="doubleQuote"/>
            <xsd:enumeration value="singleQuote"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextFields">
        <xsd:sequence>
            <xsd:element name="textField" minOccurs="1" maxOccurs="unbounded" type="CT_TextField"/>
        </xsd:sequence>
        <xsd:attribute name="count" use="optional" type="xsd:unsignedInt" default="1"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TextField">
        <xsd:attribute name="type" use="optional" type="ST_ExternalConnectionType" default="general"/>
        <xsd:attribute name="position" use="optional" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_ExternalConnectionType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="general"/>
            <xsd:enumeration value="text"/>
            <xsd:enumeration value="MDY"/>
            <xsd:enumeration value="DMY"/>
            <xsd:enumeration value="YMD"/>
            <xsd:enumeration value="MYD"/>
            <xsd:enumeration value="DYM"/>
            <xsd:enumeration value="YDM"/>
            <xsd:enumeration value="skip"/>
            <xsd:enumeration value="EMD"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="pivotCacheDefinition" type="CT_PivotCacheDefinition"/>
```



```
    <xsd:element name="pivotCacheRecords" type="CT_PivotCacheRecords"/>
    <xsd:element name="pivotTableDefinition" type="CT_pivotTableDefinition"/>
    <xsd:complexType name="CT_PivotCacheDefinition">
        <xsd:sequence>
            <xsd:element name="cacheSource" type="CT_CacheSource" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cacheFields" type="CT_CacheFields" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cacheHierarchies" minOccurs="0" type="CT_CacheHierarchies"/>
            <xsd:element name="kpis" minOccurs="0" type="CT_PCDKPIs"/>
            <xsd:element name="tupleCache" minOccurs="0" type="CT_TupleCache"/>
            <xsd:element name="calculatedItems" minOccurs="0" type="CT_CalculatedItems"/>
            <xsd:element name="calculatedMembers" type="CT_CalculatedMembers" minOccurs="0"/>
            <xsd:element name="dimensions" type="CT_Dimensions" minOccurs="0"/>
            <xsd:element name="measureGroups" type="CT_MeasureGroups" minOccurs="0"/>
            <xsd:element name="maps" type="CT_MeasureDimensionMaps" minOccurs="0"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute ref="r:id" use="optional"/>
        <xsd:attribute name="invalid" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="saveData" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="refreshOnLoad" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="optimizeMemory" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="enableRefresh" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="refreshedBy" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="refreshedDate" type="xsd:double" use="optional"/>
        <xsd:attribute name="refreshedDateIso" type="xsd:dateTime" use="optional"/>
        <xsd:attribute name="backgroundQuery" type="xsd:boolean" default="false"/>
        <xsd:attribute name="missingItemsLimit" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="createdVersion" type="xsd:unsignedByte" use="optional" default="0"/>
        <xsd:attribute name="refreshedVersion" type="xsd:unsignedByte" use="optional" default="0"/>
        <xsd:attribute name="minRefreshableVersion" type="xsd:unsignedByte" use="optional"
          default="0"/>
        <xsd:attribute name="recordCount" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="upgradeOnRefresh" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="tupleCache" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="supportSubquery" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="supportAdvancedDrill" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CacheFields">
        <xsd:sequence>
            <xsd:element name="cacheField" type="CT_CacheField" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CacheField">
        <xsd:sequence>
            <xsd:element name="sharedItems" type="CT_SharedItems" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="fieldGroup" minOccurs="0" type="CT_FieldGroup"/>
            <xsd:element name="mpMap" minOccurs="0" maxOccurs="unbounded" type="CT_X"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="caption" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="propertyName" type="s:ST_Xstring" use="optional"/>
```

```
        <xsd:attribute name="serverField" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="uniqueList" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="numFmtId" type="ST_NumFmtId" use="optional"/>
        <xsd:attribute name="formula" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="sqlType" type="xsd:int" use="optional" default="0"/>
        <xsd:attribute name="hierarchy" type="xsd:int" use="optional" default="0"/>
        <xsd:attribute name="level" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="databaseField" type="xsd:boolean" default="true"/>
        <xsd:attribute name="mappingCount" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="memberPropertyField" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CacheSource">
        <xsd:choice minOccurs="0" maxOccurs="1">
            <xsd:element name="worksheetSource" type="CT_WorksheetSource" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="consolidation" type="CT_Consolidation" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0"/>
        </xsd:choice>
        <xsd:attribute name="type" type="ST_SourceType" use="required"/>
        <xsd:attribute name="connectionId" type="xsd:unsignedInt" default="0" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SourceType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="worksheet"/>
            <xsd:enumeration value="external"/>
            <xsd:enumeration value="consolidation"/>
            <xsd:enumeration value="scenario"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_WorksheetSource">
        <xsd:attribute name="ref" type="ST_Ref" use="optional"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="sheet" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute ref="r:id" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Consolidation">
        <xsd:sequence>
            <xsd:element name="pages" type="CT_Pages" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="rangeSets" type="CT_RangeSets" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="autoPage" type="xsd:boolean" default="true" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Pages">
        <xsd:sequence>
            <xsd:element name="page" type="CT_PCDSCPage" minOccurs="1" maxOccurs="4"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PCDSCPage">
        <xsd:sequence>
            <xsd:element name="pageItem" type="CT_PageItem" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
```



```
    </xsd:complexType>
    <xsd:complexType name="CT_PageItem">
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RangeSets">
        <xsd:sequence>
            <xsd:element name="rangeSet" type="CT_RangeSet" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RangeSet">
        <xsd:attribute name="i1" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="i2" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="i3" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="i4" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="ref" type="ST_Ref" use="optional"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="sheet" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute ref="r:id" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SharedItems">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="m" type="CT_Missing" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="n" type="CT_Number" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="b" type="CT_Boolean" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="e" type="CT_Error" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="s" type="CT_String" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="d" type="CT_DateTime" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
        <xsd:attribute name="containsSemiMixedTypes" type="xsd:boolean" use="optional"
          default="true"/>
        <xsd:attribute name="containsNonDate" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="containsDate" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="containsString" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="containsBlank" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="containsMixedTypes" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="containsNumber" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="containsInteger" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="minValue" type="xsd:double" use="optional"/>
        <xsd:attribute name="maxValue" type="xsd:double" use="optional"/>
        <xsd:attribute name="minDate" type="xsd:dateTime" use="optional"/>
        <xsd:attribute name="maxDate" type="xsd:dateTime" use="optional"/>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="longText" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Missing">
        <xsd:sequence>
            <xsd:element name="tpls" minOccurs="0" maxOccurs="unbounded" type="CT_Tuples"/>
            <xsd:element name="x" minOccurs="0" maxOccurs="unbounded" type="CT_X"/>
        </xsd:sequence>
        <xsd:attribute name="u" type="xsd:boolean"/>
        <xsd:attribute name="f" type="xsd:boolean"/>
        <xsd:attribute name="c" type="s:ST_Xstring"/>
```

```
        <xsd:attribute name="cp" type="xsd:unsignedInt"/>
        <xsd:attribute name="in" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="bc" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="fc" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="i" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="un" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="st" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="b" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Number">
        <xsd:sequence>
            <xsd:element name="tpls" minOccurs="0" maxOccurs="unbounded" type="CT_Tuples"/>
            <xsd:element name="x" minOccurs="0" maxOccurs="unbounded" type="CT_X"/>
        </xsd:sequence>
        <xsd:attribute name="v" use="required" type="xsd:double"/>
        <xsd:attribute name="u" type="xsd:boolean"/>
        <xsd:attribute name="f" type="xsd:boolean"/>
        <xsd:attribute name="c" type="s:ST_Xstring"/>
        <xsd:attribute name="cp" type="xsd:unsignedInt"/>
        <xsd:attribute name="in" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="bc" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="fc" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="i" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="un" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="st" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="b" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Boolean">
        <xsd:sequence>
            <xsd:element name="x" minOccurs="0" maxOccurs="unbounded" type="CT_X"/>
        </xsd:sequence>
        <xsd:attribute name="v" use="required" type="xsd:boolean"/>
        <xsd:attribute name="u" type="xsd:boolean"/>
        <xsd:attribute name="f" type="xsd:boolean"/>
        <xsd:attribute name="c" type="s:ST_Xstring"/>
        <xsd:attribute name="cp" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Error">
        <xsd:sequence>
            <xsd:element name="tpls" minOccurs="0" type="CT_Tuples"/>
            <xsd:element name="x" minOccurs="0" maxOccurs="unbounded" type="CT_X"/>
        </xsd:sequence>
        <xsd:attribute name="v" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="u" type="xsd:boolean"/>
        <xsd:attribute name="f" type="xsd:boolean"/>
        <xsd:attribute name="c" type="s:ST_Xstring"/>
        <xsd:attribute name="cp" type="xsd:unsignedInt"/>
        <xsd:attribute name="in" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="bc" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="fc" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="i" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="un" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="st" type="xsd:boolean" use="optional" default="false"/>
```



```
        <xsd:attribute name="b" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_String">
        <xsd:sequence>
            <xsd:element name="tpls" minOccurs="0" maxOccurs="unbounded" type="CT_Tuples"/>
            <xsd:element name="x" minOccurs="0" maxOccurs="unbounded" type="CT_X"/>
        </xsd:sequence>
        <xsd:attribute name="v" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="u" type="xsd:boolean"/>
        <xsd:attribute name="f" type="xsd:boolean"/>
        <xsd:attribute name="c" type="s:ST_Xstring"/>
        <xsd:attribute name="cp" type="xsd:unsignedInt"/>
        <xsd:attribute name="in" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="bc" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="fc" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="i" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="un" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="st" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="b" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DateTime">
        <xsd:sequence>
            <xsd:element name="x" minOccurs="0" maxOccurs="unbounded" type="CT_X"/>
        </xsd:sequence>
        <xsd:attribute name="v" use="required" type="xsd:dateTime"/>
        <xsd:attribute name="u" type="xsd:boolean"/>
        <xsd:attribute name="f" type="xsd:boolean"/>
        <xsd:attribute name="c" type="s:ST_Xstring"/>
        <xsd:attribute name="cp" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FieldGroup">
        <xsd:sequence>
            <xsd:element name="rangePr" minOccurs="0" type="CT_RangePr"/>
            <xsd:element name="discretePr" minOccurs="0" type="CT_DiscretePr"/>
            <xsd:element name="groupItems" minOccurs="0" type="CT_GroupItems"/>
        </xsd:sequence>
        <xsd:attribute name="par" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="base" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RangePr">
        <xsd:attribute name="autoStart" type="xsd:boolean" default="true"/>
        <xsd:attribute name="autoEnd" type="xsd:boolean" default="true"/>
        <xsd:attribute name="groupBy" type="ST_GroupBy" default="range"/>
        <xsd:attribute name="startNum" type="xsd:double"/>
        <xsd:attribute name="endNum" type="xsd:double"/>
        <xsd:attribute name="startDate" type="xsd:dateTime"/>
        <xsd:attribute name="endDate" type="xsd:dateTime"/>
        <xsd:attribute name="groupInterval" type="xsd:double" default="1"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_GroupBy">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="range"/>
            <xsd:enumeration value="seconds"/>
```

```
            <xsd:enumeration value="minutes"/>
            <xsd:enumeration value="hours"/>
            <xsd:enumeration value="days"/>
            <xsd:enumeration value="months"/>
            <xsd:enumeration value="quarters"/>
            <xsd:enumeration value="years"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DiscretePr">
        <xsd:sequence>
            <xsd:element name="x" maxOccurs="unbounded" type="CT_Index"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupItems">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="m" type="CT_Missing"/>
            <xsd:element name="n" type="CT_Number"/>
            <xsd:element name="b" type="CT_Boolean"/>
            <xsd:element name="e" type="CT_Error"/>
            <xsd:element name="s" type="CT_String"/>
            <xsd:element name="d" type="CT_DateTime"/>
        </xsd:choice>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotCacheRecords">
        <xsd:sequence>
            <xsd:element name="r" minOccurs="0" maxOccurs="unbounded" type="CT_Record"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Record">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="m" type="CT_Missing"/>
            <xsd:element name="n" type="CT_Number"/>
            <xsd:element name="b" type="CT_Boolean"/>
            <xsd:element name="e" type="CT_Error"/>
            <xsd:element name="s" type="CT_String"/>
            <xsd:element name="d" type="CT_DateTime"/>
            <xsd:element name="x" type="CT_Index"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_PCDKPIs">
        <xsd:sequence>
            <xsd:element name="kpi" minOccurs="0" maxOccurs="unbounded" type="CT_PCDKPI"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PCDKPI">
        <xsd:attribute name="uniqueName" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="caption" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="displayFolder" type="s:ST_Xstring"/>
```



```
        <xsd:attribute name="measureGroup" type="s:ST_Xstring"/>
        <xsd:attribute name="parent" type="s:ST_Xstring"/>
        <xsd:attribute name="value" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="goal" type="s:ST_Xstring"/>
        <xsd:attribute name="status" type="s:ST_Xstring"/>
        <xsd:attribute name="trend" type="s:ST_Xstring"/>
        <xsd:attribute name="weight" type="s:ST_Xstring"/>
        <xsd:attribute name="time" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CacheHierarchies">
        <xsd:sequence>
            <xsd:element name="cacheHierarchy" minOccurs="0" maxOccurs="unbounded"
              type="CT_CacheHierarchy"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CacheHierarchy">
        <xsd:sequence>
            <xsd:element name="fieldsUsage" minOccurs="0" type="CT_FieldsUsage"/>
            <xsd:element name="groupLevels" minOccurs="0" type="CT_GroupLevels"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="uniqueName" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="caption" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="measure" type="xsd:boolean" default="false"/>
        <xsd:attribute name="set" type="xsd:boolean" default="false"/>
        <xsd:attribute name="parentSet" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="iconSet" type="xsd:int" default="0"/>
        <xsd:attribute name="attribute" type="xsd:boolean" default="false"/>
        <xsd:attribute name="time" type="xsd:boolean" default="false"/>
        <xsd:attribute name="keyAttribute" type="xsd:boolean" default="false"/>
        <xsd:attribute name="defaultMemberUniqueName" type="s:ST_Xstring"/>
        <xsd:attribute name="allUniqueName" type="s:ST_Xstring"/>
        <xsd:attribute name="allCaption" type="s:ST_Xstring"/>
        <xsd:attribute name="dimensionUniqueName" type="s:ST_Xstring"/>
        <xsd:attribute name="displayFolder" type="s:ST_Xstring"/>
        <xsd:attribute name="measureGroup" type="s:ST_Xstring"/>
        <xsd:attribute name="measures" type="xsd:boolean" default="false"/>
        <xsd:attribute name="count" use="required" type="xsd:unsignedInt"/>
        <xsd:attribute name="oneField" type="xsd:boolean" default="false"/>
        <xsd:attribute name="memberValueDatatype" use="optional" type="xsd:unsignedShort"/>
        <xsd:attribute name="unbalanced" use="optional" type="xsd:boolean"/>
        <xsd:attribute name="unbalancedGroup" use="optional" type="xsd:boolean"/>
        <xsd:attribute name="hidden" type="xsd:boolean" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FieldsUsage">
        <xsd:sequence>
            <xsd:element name="fieldUsage" minOccurs="0" maxOccurs="unbounded" type="CT_FieldUsage"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FieldUsage">
        <xsd:attribute name="x" use="required" type="xsd:int"/>
```

```
    </xsd:complexType>
    <xsd:complexType name="CT_GroupLevels">
        <xsd:sequence>
            <xsd:element name="groupLevel" maxOccurs="unbounded" type="CT_GroupLevel"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupLevel">
        <xsd:sequence>
            <xsd:element name="groups" minOccurs="0" type="CT_Groups"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="uniqueName" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="caption" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="user" type="xsd:boolean" default="false"/>
        <xsd:attribute name="customRollUp" type="xsd:boolean" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Groups">
        <xsd:sequence>
            <xsd:element name="group" maxOccurs="unbounded" type="CT_LevelGroup"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_LevelGroup">
        <xsd:sequence>
            <xsd:element name="groupMembers" type="CT_GroupMembers"/>
        </xsd:sequence>
        <xsd:attribute name="name" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="uniqueName" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="caption" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="uniqueParent" type="s:ST_Xstring"/>
        <xsd:attribute name="id" type="xsd:int"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupMembers">
        <xsd:sequence>
            <xsd:element name="groupMember" maxOccurs="unbounded" type="CT_GroupMember"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupMember">
        <xsd:attribute name="uniqueName" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="group" type="xsd:boolean" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TupleCache">
        <xsd:sequence>
            <xsd:element name="entries" minOccurs="0" type="CT_PCDSDTCEntries"/>
            <xsd:element name="sets" minOccurs="0" type="CT_Sets"/>
            <xsd:element name="queryCache" minOccurs="0" type="CT_QueryCache"/>
            <xsd:element name="serverFormats" minOccurs="0" maxOccurs="1" type="CT_ServerFormats"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ServerFormat">
```



```
        <xsd:attribute name="culture" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="format" use="optional" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ServerFormats">
        <xsd:sequence>
            <xsd:element name="serverFormat" type="CT_ServerFormat" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PCDSDTCEntries">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="m" type="CT_Missing"/>
            <xsd:element name="n" type="CT_Number"/>
            <xsd:element name="e" type="CT_Error"/>
            <xsd:element name="s" type="CT_String"/>
        </xsd:choice>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Tuples">
        <xsd:sequence>
            <xsd:element name="tpl" type="CT_Tuple" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="c" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Tuple">
        <xsd:attribute name="fld" type="xsd:unsignedInt"/>
        <xsd:attribute name="hier" type="xsd:unsignedInt"/>
        <xsd:attribute name="item" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Sets">
        <xsd:sequence>
            <xsd:element name="set" maxOccurs="unbounded" type="CT_Set"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Set">
        <xsd:sequence>
            <xsd:element name="tpls" minOccurs="0" maxOccurs="unbounded" type="CT_Tuples"/>
            <xsd:element name="sortByTuple" minOccurs="0" type="CT_Tuples"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
        <xsd:attribute name="maxRank" use="required" type="xsd:int"/>
        <xsd:attribute name="setDefinition" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="sortType" type="ST_SortType" default="none"/>
        <xsd:attribute name="queryFailed" type="xsd:boolean" default="false"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SortType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="ascending"/>
            <xsd:enumeration value="descending"/>
            <xsd:enumeration value="ascendingAlpha"/>
```

```
            <xsd:enumeration value="descendingAlpha"/>
            <xsd:enumeration value="ascendingNatural"/>
            <xsd:enumeration value="descendingNatural"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_QueryCache">
        <xsd:sequence>
            <xsd:element name="query" maxOccurs="unbounded" type="CT_Query"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Query">
        <xsd:sequence>
            <xsd:element name="tpls" minOccurs="0" type="CT_Tuples"/>
        </xsd:sequence>
        <xsd:attribute name="mdx" use="required" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CalculatedItems">
        <xsd:sequence>
            <xsd:element name="calculatedItem" maxOccurs="unbounded" type="CT_CalculatedItem"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CalculatedItem">
        <xsd:sequence>
            <xsd:element name="pivotArea" type="CT_PivotArea"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="field" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="formula" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CalculatedMembers">
        <xsd:sequence>
            <xsd:element name="calculatedMember" maxOccurs="unbounded" type="CT_CalculatedMember"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CalculatedMember">
        <xsd:sequence minOccurs="0">
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="name" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="mdx" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="memberName" type="s:ST_Xstring"/>
        <xsd:attribute name="hierarchy" type="s:ST_Xstring"/>
        <xsd:attribute name="parent" type="s:ST_Xstring"/>
        <xsd:attribute name="solveOrder" type="xsd:int" default="0"/>
        <xsd:attribute name="set" type="xsd:boolean" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_pivotTableDefinition">
        <xsd:sequence>
            <xsd:element name="location" type="CT_Location"/>
            <xsd:element name="pivotFields" type="CT_PivotFields" minOccurs="0"/>
```



```
            <xsd:element name="rowFields" type="CT_RowFields" minOccurs="0"/>
            <xsd:element name="rowItems" type="CT_rowItems" minOccurs="0"/>
            <xsd:element name="colFields" type="CT_ColFields" minOccurs="0"/>
            <xsd:element name="colItems" type="CT_colItems" minOccurs="0"/>
            <xsd:element name="pageFields" type="CT_PageFields" minOccurs="0"/>
            <xsd:element name="dataFields" type="CT_DataFields" minOccurs="0"/>
            <xsd:element name="formats" type="CT_Formats" minOccurs="0"/>
            <xsd:element name="conditionalFormats" type="CT_ConditionalFormats" minOccurs="0"/>
            <xsd:element name="chartFormats" type="CT_ChartFormats" minOccurs="0"/>
            <xsd:element name="pivotHierarchies" type="CT_PivotHierarchies" minOccurs="0"/>
            <xsd:element name="pivotTableStyleInfo" minOccurs="0" maxOccurs="1"
              type="CT_PivotTableStyle"/>
            <xsd:element name="filters" minOccurs="0" maxOccurs="1" type="CT_PivotFilters"/>
            <xsd:element name="rowHierarchiesUsage" type="CT_RowHierarchiesUsage" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="colHierarchiesUsage" type="CT_ColHierarchiesUsage" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="name" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="cacheId" use="required" type="xsd:unsignedInt"/>
        <xsd:attribute name="dataOnRows" type="xsd:boolean" default="false"/>
        <xsd:attribute name="dataPosition" type="xsd:unsignedInt" use="optional"/>
        <xsd:attributeGroup ref="AG_AutoFormat"/>
        <xsd:attribute name="dataCaption" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="grandTotalCaption" type="s:ST_Xstring"/>
        <xsd:attribute name="errorCaption" type="s:ST_Xstring"/>
        <xsd:attribute name="showError" type="xsd:boolean" default="false"/>
        <xsd:attribute name="missingCaption" type="s:ST_Xstring"/>
        <xsd:attribute name="showMissing" type="xsd:boolean" default="true"/>
        <xsd:attribute name="pageStyle" type="s:ST_Xstring"/>
        <xsd:attribute name="pivotTableStyle" type="s:ST_Xstring"/>
        <xsd:attribute name="vacatedStyle" type="s:ST_Xstring"/>
        <xsd:attribute name="tag" type="s:ST_Xstring"/>
        <xsd:attribute name="updatedVersion" type="xsd:unsignedByte" default="0"/>
        <xsd:attribute name="minRefreshableVersion" type="xsd:unsignedByte" default="0"/>
        <xsd:attribute name="asteriskTotals" type="xsd:boolean" default="false"/>
        <xsd:attribute name="showItems" type="xsd:boolean" default="true"/>
        <xsd:attribute name="editData" type="xsd:boolean" default="false"/>
        <xsd:attribute name="disableFieldList" type="xsd:boolean" default="false"/>
        <xsd:attribute name="showCalcMbrs" type="xsd:boolean" default="true"/>
        <xsd:attribute name="visualTotals" type="xsd:boolean" default="true"/>
        <xsd:attribute name="showMultipleLabel" type="xsd:boolean" default="true"/>
        <xsd:attribute name="showDataDropDown" type="xsd:boolean" default="true"/>
        <xsd:attribute name="showDrill" type="xsd:boolean" default="true"/>
        <xsd:attribute name="printDrill" type="xsd:boolean" default="false"/>
        <xsd:attribute name="showMemberPropertyTips" type="xsd:boolean" default="true"/>
        <xsd:attribute name="showDataTips" type="xsd:boolean" default="true"/>
        <xsd:attribute name="enableWizard" type="xsd:boolean" default="true"/>
        <xsd:attribute name="enableDrill" type="xsd:boolean" default="true"/>
        <xsd:attribute name="enableFieldProperties" type="xsd:boolean" default="true"/>
        <xsd:attribute name="preserveFormatting" type="xsd:boolean" default="true"/>
        <xsd:attribute name="useAutoFormatting" type="xsd:boolean" default="false"/>
```

```
        <xsd:attribute name="pageWrap" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="pageOverThenDown" type="xsd:boolean" default="false"/>
        <xsd:attribute name="subtotalHiddenItems" type="xsd:boolean" default="false"/>
        <xsd:attribute name="rowGrandTotals" type="xsd:boolean" default="true"/>
        <xsd:attribute name="colGrandTotals" type="xsd:boolean" default="true"/>
        <xsd:attribute name="fieldPrintTitles" type="xsd:boolean" default="false"/>
        <xsd:attribute name="itemPrintTitles" type="xsd:boolean" default="false"/>
        <xsd:attribute name="mergeItem" type="xsd:boolean" default="false"/>
        <xsd:attribute name="showDropZones" type="xsd:boolean" default="true"/>
        <xsd:attribute name="createdVersion" type="xsd:unsignedByte" default="0"/>
        <xsd:attribute name="indent" type="xsd:unsignedInt" default="1"/>
        <xsd:attribute name="showEmptyRow" type="xsd:boolean" default="false"/>
        <xsd:attribute name="showEmptyCol" type="xsd:boolean" default="false"/>
        <xsd:attribute name="showHeaders" type="xsd:boolean" default="true"/>
        <xsd:attribute name="compact" type="xsd:boolean" default="true"/>
        <xsd:attribute name="outline" type="xsd:boolean" default="false"/>
        <xsd:attribute name="outlineData" type="xsd:boolean" default="false"/>
        <xsd:attribute name="compactData" type="xsd:boolean" default="true"/>
        <xsd:attribute name="published" type="xsd:boolean" default="false"/>
        <xsd:attribute name="gridDropZones" type="xsd:boolean" default="false"/>
        <xsd:attribute name="immersive" type="xsd:boolean" default="true"/>
        <xsd:attribute name="multipleFieldFilters" type="xsd:boolean" default="true"/>
        <xsd:attribute name="chartFormat" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="rowHeaderCaption" type="s:ST_Xstring"/>
        <xsd:attribute name="colHeaderCaption" type="s:ST_Xstring"/>
        <xsd:attribute name="fieldListSortAscending" type="xsd:boolean" default="false"/>
        <xsd:attribute name="mdxSubqueries" type="xsd:boolean" default="false"/>
        <xsd:attribute name="customListSort" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Location">
        <xsd:attribute name="ref" use="required" type="ST_Ref"/>
        <xsd:attribute name="firstHeaderRow" use="required" type="xsd:unsignedInt"/>
        <xsd:attribute name="firstDataRow" use="required" type="xsd:unsignedInt"/>
        <xsd:attribute name="firstDataCol" use="required" type="xsd:unsignedInt"/>
        <xsd:attribute name="rowPageCount" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="colPageCount" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotFields">
        <xsd:sequence>
            <xsd:element name="pivotField" maxOccurs="unbounded" type="CT_PivotField"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotField">
        <xsd:sequence>
            <xsd:element name="items" minOccurs="0" type="CT_Items"/>
            <xsd:element name="autoSortScope" minOccurs="0" type="CT_AutoSortScope"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="s:ST_Xstring"/>
        <xsd:attribute name="axis" use="optional" type="ST_Axis"/>
        <xsd:attribute name="dataField" type="xsd:boolean" default="false"/>
        <xsd:attribute name="subtotalCaption" type="s:ST_Xstring"/>
```



```
        <xsd:attribute name="showDropDowns" type="xsd:boolean" default="true"/>
        <xsd:attribute name="hiddenLevel" type="xsd:boolean" default="false"/>
        <xsd:attribute name="uniqueMemberProperty" type="s:ST_Xstring"/>
        <xsd:attribute name="compact" type="xsd:boolean" default="true"/>
        <xsd:attribute name="allDrilled" type="xsd:boolean" default="false"/>
        <xsd:attribute name="numFmtId" type="ST_NumFmtId" use="optional"/>
        <xsd:attribute name="outline" type="xsd:boolean" default="true"/>
        <xsd:attribute name="subtotalTop" type="xsd:boolean" default="true"/>
        <xsd:attribute name="dragToRow" type="xsd:boolean" default="true"/>
        <xsd:attribute name="dragToCol" type="xsd:boolean" default="true"/>
        <xsd:attribute name="multipleItemSelectionAllowed" type="xsd:boolean" default="false"/>
        <xsd:attribute name="dragToPage" type="xsd:boolean" default="true"/>
        <xsd:attribute name="dragToData" type="xsd:boolean" default="true"/>
        <xsd:attribute name="dragOff" type="xsd:boolean" default="true"/>
        <xsd:attribute name="showAll" type="xsd:boolean" default="true"/>
        <xsd:attribute name="insertBlankRow" type="xsd:boolean" default="false"/>
        <xsd:attribute name="serverField" type="xsd:boolean" default="false"/>
        <xsd:attribute name="insertPageBreak" type="xsd:boolean" default="false"/>
        <xsd:attribute name="autoShow" type="xsd:boolean" default="false"/>
        <xsd:attribute name="topAutoShow" type="xsd:boolean" default="true"/>
        <xsd:attribute name="hideNewItems" type="xsd:boolean" default="false"/>
        <xsd:attribute name="measureFilter" type="xsd:boolean" default="false"/>
        <xsd:attribute name="includeNewItemsInFilter" type="xsd:boolean" default="false"/>
        <xsd:attribute name="itemPageCount" type="xsd:unsignedInt" default="10"/>
        <xsd:attribute name="sortType" type="ST_FieldSortType" default="manual"/>
        <xsd:attribute name="dataSourceSort" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="nonAutoSortDefault" type="xsd:boolean" default="false"/>
        <xsd:attribute name="rankBy" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="defaultSubtotal" type="xsd:boolean" default="true"/>
        <xsd:attribute name="sumSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="countASubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="avgSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="maxSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="minSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="productSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="countSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="stdDevSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="stdDevPSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="varSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="varPSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="showPropCell" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showPropTip" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showPropAsCaption" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="defaultAttributeDrillState" type="xsd:boolean" use="optional"
          default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AutoSortScope">
        <xsd:sequence>
            <xsd:element name="pivotArea" type="CT_PivotArea"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Items">
        <xsd:sequence>
```

```
            <xsd:element name="item" maxOccurs="unbounded" type="CT_Item"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Item">
        <xsd:attribute name="n" type="s:ST_Xstring"/>
        <xsd:attribute name="t" type="ST_ItemType" default="data"/>
        <xsd:attribute name="h" type="xsd:boolean" default="false"/>
        <xsd:attribute name="s" type="xsd:boolean" default="false"/>
        <xsd:attribute name="sd" type="xsd:boolean" default="true"/>
        <xsd:attribute name="f" type="xsd:boolean" default="false"/>
        <xsd:attribute name="m" type="xsd:boolean" default="false"/>
        <xsd:attribute name="c" type="xsd:boolean" default="false"/>
        <xsd:attribute name="x" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="d" type="xsd:boolean" default="false"/>
        <xsd:attribute name="e" type="xsd:boolean" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PageFields">
        <xsd:sequence>
            <xsd:element name="pageField" maxOccurs="unbounded" type="CT_PageField"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PageField">
        <xsd:sequence minOccurs="0">
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="fld" use="required" type="xsd:int"/>
        <xsd:attribute name="item" use="optional" type="xsd:unsignedInt"/>
        <xsd:attribute name="hier" type="xsd:int"/>
        <xsd:attribute name="name" type="s:ST_Xstring"/>
        <xsd:attribute name="cap" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DataFields">
        <xsd:sequence>
            <xsd:element name="dataField" maxOccurs="unbounded" type="CT_DataField"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DataField">
        <xsd:sequence>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="name" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="fld" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="subtotal" type="ST_DataConsolidateFunction" default="sum"/>
        <xsd:attribute name="showDataAs" type="ST_ShowDataAs" default="normal"/>
        <xsd:attribute name="baseField" type="xsd:int" default="-1"/>
        <xsd:attribute name="baseItem" type="xsd:unsignedInt" default="1048832"/>
        <xsd:attribute name="numFmtId" type="ST_NumFmtId" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_rowItems">
        <xsd:sequence>
```



```
            <xsd:element name="i" maxOccurs="unbounded" type="CT_I"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_colItems">
        <xsd:sequence>
            <xsd:element name="i" maxOccurs="unbounded" type="CT_I"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_I">
        <xsd:sequence>
            <xsd:element name="x" minOccurs="0" maxOccurs="unbounded" type="CT_X"/>
        </xsd:sequence>
        <xsd:attribute name="t" type="ST_ItemType" default="data"/>
        <xsd:attribute name="r" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="i" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_X">
        <xsd:attribute name="v" type="xsd:int" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RowFields">
        <xsd:sequence>
            <xsd:element name="field" maxOccurs="unbounded" type="CT_Field"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ColFields">
        <xsd:sequence>
            <xsd:element name="field" maxOccurs="unbounded" type="CT_Field"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Field">
        <xsd:attribute name="x" type="xsd:int" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Formats">
        <xsd:sequence>
            <xsd:element name="format" maxOccurs="unbounded" type="CT_Format"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Format">
        <xsd:sequence>
            <xsd:element name="pivotArea" type="CT_PivotArea"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="action" type="ST_FormatAction" default="formatting"/>
        <xsd:attribute name="dxfId" type="ST_DxfId" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ConditionalFormats">
        <xsd:sequence>
            <xsd:element name="conditionalFormat" maxOccurs="unbounded" type="CT_ConditionalFormat"/>
```

```
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ConditionalFormat">
        <xsd:sequence>
            <xsd:element name="pivotAreas" type="CT_PivotAreas"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="scope" type="ST_Scope" default="selection"/>
        <xsd:attribute name="type" type="ST_Type" default="none"/>
        <xsd:attribute name="priority" use="required" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotAreas">
        <xsd:sequence>
            <xsd:element name="pivotArea" minOccurs="0" maxOccurs="unbounded" type="CT_PivotArea"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Scope">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="selection"/>
            <xsd:enumeration value="data"/>
            <xsd:enumeration value="field"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Type">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="all"/>
            <xsd:enumeration value="row"/>
            <xsd:enumeration value="column"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_ChartFormats">
        <xsd:sequence>
            <xsd:element name="chartFormat" maxOccurs="unbounded" type="CT_ChartFormat"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ChartFormat">
        <xsd:sequence>
            <xsd:element name="pivotArea" type="CT_PivotArea"/>
        </xsd:sequence>
        <xsd:attribute name="chart" use="required" type="xsd:unsignedInt"/>
        <xsd:attribute name="format" use="required" type="xsd:unsignedInt"/>
        <xsd:attribute name="series" type="xsd:boolean" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotHierarchies">
        <xsd:sequence>
            <xsd:element name="pivotHierarchy" maxOccurs="unbounded" type="CT_PivotHierarchy"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
```



```
    <xsd:complexType name="CT_PivotHierarchy">
        <xsd:sequence>
            <xsd:element name="mps" minOccurs="0" type="CT_MemberProperties"/>
            <xsd:element name="members" minOccurs="0" maxOccurs="unbounded" type="CT_Members"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="outline" type="xsd:boolean" default="false"/>
        <xsd:attribute name="multipleItemSelectionAllowed" type="xsd:boolean" default="false"/>
        <xsd:attribute name="subtotalTop" type="xsd:boolean" default="false"/>
        <xsd:attribute name="showInFieldList" type="xsd:boolean" default="true"/>
        <xsd:attribute name="dragToRow" type="xsd:boolean" default="true"/>
        <xsd:attribute name="dragToCol" type="xsd:boolean" default="true"/>
        <xsd:attribute name="dragToPage" type="xsd:boolean" default="true"/>
        <xsd:attribute name="dragToData" type="xsd:boolean" default="false"/>
        <xsd:attribute name="dragOff" type="xsd:boolean" default="true"/>
        <xsd:attribute name="includeNewItemsInFilter" type="xsd:boolean" default="false"/>
        <xsd:attribute name="caption" type="s:ST_Xstring" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RowHierarchiesUsage">
        <xsd:sequence>
            <xsd:element name="rowHierarchyUsage" minOccurs="1" maxOccurs="unbounded"
              type="CT_HierarchyUsage"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ColHierarchiesUsage">
        <xsd:sequence>
            <xsd:element name="colHierarchyUsage" minOccurs="1" maxOccurs="unbounded"
              type="CT_HierarchyUsage"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_HierarchyUsage">
        <xsd:attribute name="hierarchyUsage" type="xsd:int" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MemberProperties">
        <xsd:sequence>
            <xsd:element name="mp" maxOccurs="unbounded" type="CT_MemberProperty"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MemberProperty">
        <xsd:attribute name="name" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="showCell" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showTip" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showAsCaption" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="nameLen" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="pPos" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="pLen" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="level" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="field" use="required" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Members">
```

```
        <xsd:sequence>
            <xsd:element name="member" maxOccurs="unbounded" type="CT_Member"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
        <xsd:attribute name="level" use="optional" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Member">
        <xsd:attribute name="name" use="required" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Dimensions">
        <xsd:sequence>
            <xsd:element name="dimension" minOccurs="0" maxOccurs="unbounded"
              type="CT_PivotDimension"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotDimension">
        <xsd:attribute name="measure" type="xsd:boolean" default="false"/>
        <xsd:attribute name="name" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="uniqueName" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="caption" use="required" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MeasureGroups">
        <xsd:sequence>
            <xsd:element name="measureGroup" minOccurs="0" maxOccurs="unbounded"
              type="CT_MeasureGroup"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MeasureDimensionMaps">
        <xsd:sequence>
            <xsd:element name="map" minOccurs="0" maxOccurs="unbounded"
              type="CT_MeasureDimensionMap"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MeasureGroup">
        <xsd:attribute name="name" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="caption" use="required" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MeasureDimensionMap">
        <xsd:attribute name="measureGroup" use="optional" type="xsd:unsignedInt"/>
        <xsd:attribute name="dimension" use="optional" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotTableStyle">
        <xsd:attribute name="name" type="xsd:string"/>
        <xsd:attribute name="showRowHeaders" type="xsd:boolean"/>
        <xsd:attribute name="showColHeaders" type="xsd:boolean"/>
        <xsd:attribute name="showRowStripes" type="xsd:boolean"/>
        <xsd:attribute name="showColStripes" type="xsd:boolean"/>
        <xsd:attribute name="showLastColumn" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotFilters">
```



```
        <xsd:sequence>
            <xsd:element name="filter" minOccurs="0" maxOccurs="unbounded" type="CT_PivotFilter"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotFilter">
        <xsd:sequence>
            <xsd:element name="autoFilter" minOccurs="1" maxOccurs="1" type="CT_AutoFilter"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="fld" use="required" type="xsd:unsignedInt"/>
        <xsd:attribute name="mpFld" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="type" use="required" type="ST_PivotFilterType"/>
        <xsd:attribute name="evalOrder" use="optional" type="xsd:int" default="0"/>
        <xsd:attribute name="id" use="required" type="xsd:unsignedInt"/>
        <xsd:attribute name="iMeasureHier" use="optional" type="xsd:unsignedInt"/>
        <xsd:attribute name="iMeasureFld" use="optional" type="xsd:unsignedInt"/>
        <xsd:attribute name="name" type="s:ST_Xstring"/>
        <xsd:attribute name="description" type="s:ST_Xstring"/>
        <xsd:attribute name="stringValue1" type="s:ST_Xstring"/>
        <xsd:attribute name="stringValue2" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_ShowDataAs">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="normal"/>
            <xsd:enumeration value="difference"/>
            <xsd:enumeration value="percent"/>
            <xsd:enumeration value="percentDiff"/>
            <xsd:enumeration value="runTotal"/>
            <xsd:enumeration value="percentOfRow"/>
            <xsd:enumeration value="percentOfCol"/>
            <xsd:enumeration value="percentOfTotal"/>
            <xsd:enumeration value="index"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ItemType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="data"/>
            <xsd:enumeration value="default"/>
            <xsd:enumeration value="sum"/>
            <xsd:enumeration value="countA"/>
            <xsd:enumeration value="avg"/>
            <xsd:enumeration value="max"/>
            <xsd:enumeration value="min"/>
            <xsd:enumeration value="product"/>
            <xsd:enumeration value="count"/>
            <xsd:enumeration value="stdDev"/>
            <xsd:enumeration value="stdDevP"/>
            <xsd:enumeration value="var"/>
            <xsd:enumeration value="varP"/>
            <xsd:enumeration value="grand"/>
            <xsd:enumeration value="blank"/>
        </xsd:restriction>
```

```
    </xsd:simpleType>
    <xsd:simpleType name="ST_FormatAction">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="blank"/>
            <xsd:enumeration value="formatting"/>
            <xsd:enumeration value="drill"/>
            <xsd:enumeration value="formula"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FieldSortType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="manual"/>
            <xsd:enumeration value="ascending"/>
            <xsd:enumeration value="descending"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PivotFilterType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="unknown"/>
            <xsd:enumeration value="count"/>
            <xsd:enumeration value="percent"/>
            <xsd:enumeration value="sum"/>
            <xsd:enumeration value="captionEqual"/>
            <xsd:enumeration value="captionNotEqual"/>
            <xsd:enumeration value="captionBeginsWith"/>
            <xsd:enumeration value="captionNotBeginsWith"/>
            <xsd:enumeration value="captionEndsWith"/>
            <xsd:enumeration value="captionNotEndsWith"/>
            <xsd:enumeration value="captionContains"/>
            <xsd:enumeration value="captionNotContains"/>
            <xsd:enumeration value="captionGreaterThan"/>
            <xsd:enumeration value="captionGreaterThanOrEqual"/>
            <xsd:enumeration value="captionLessThan"/>
            <xsd:enumeration value="captionLessThanOrEqual"/>
            <xsd:enumeration value="captionBetween"/>
            <xsd:enumeration value="captionNotBetween"/>
            <xsd:enumeration value="valueEqual"/>
            <xsd:enumeration value="valueNotEqual"/>
            <xsd:enumeration value="valueGreaterThan"/>
            <xsd:enumeration value="valueGreaterThanOrEqual"/>
            <xsd:enumeration value="valueLessThan"/>
            <xsd:enumeration value="valueLessThanOrEqual"/>
            <xsd:enumeration value="valueBetween"/>
            <xsd:enumeration value="valueNotBetween"/>
            <xsd:enumeration value="dateEqual"/>
            <xsd:enumeration value="dateNotEqual"/>
            <xsd:enumeration value="dateOlderThan"/>
            <xsd:enumeration value="dateOlderThanOrEqual"/>
            <xsd:enumeration value="dateNewerThan"/>
            <xsd:enumeration value="dateNewerThanOrEqual"/>
            <xsd:enumeration value="dateBetween"/>
            <xsd:enumeration value="dateNotBetween"/>
            <xsd:enumeration value="tomorrow"/>
```



```
            <xsd:enumeration value="today"/>
            <xsd:enumeration value="yesterday"/>
            <xsd:enumeration value="nextWeek"/>
            <xsd:enumeration value="thisWeek"/>
            <xsd:enumeration value="lastWeek"/>
            <xsd:enumeration value="nextMonth"/>
            <xsd:enumeration value="thisMonth"/>
            <xsd:enumeration value="lastMonth"/>
            <xsd:enumeration value="nextQuarter"/>
            <xsd:enumeration value="thisQuarter"/>
            <xsd:enumeration value="lastQuarter"/>
            <xsd:enumeration value="nextYear"/>
            <xsd:enumeration value="thisYear"/>
            <xsd:enumeration value="lastYear"/>
            <xsd:enumeration value="yearToDate"/>
            <xsd:enumeration value="Q1"/>
            <xsd:enumeration value="Q2"/>
            <xsd:enumeration value="Q3"/>
            <xsd:enumeration value="Q4"/>
            <xsd:enumeration value="M1"/>
            <xsd:enumeration value="M2"/>
            <xsd:enumeration value="M3"/>
            <xsd:enumeration value="M4"/>
            <xsd:enumeration value="M5"/>
            <xsd:enumeration value="M6"/>
            <xsd:enumeration value="M7"/>
            <xsd:enumeration value="M8"/>
            <xsd:enumeration value="M9"/>
            <xsd:enumeration value="M10"/>
            <xsd:enumeration value="M11"/>
            <xsd:enumeration value="M12"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PivotArea">
        <xsd:sequence>
            <xsd:element name="references" minOccurs="0" type="CT_PivotAreaReferences"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="field" use="optional" type="xsd:int"/>
        <xsd:attribute name="type" type="ST_PivotAreaType" default="normal"/>
        <xsd:attribute name="dataOnly" type="xsd:boolean" default="true"/>
        <xsd:attribute name="labelOnly" type="xsd:boolean" default="false"/>
        <xsd:attribute name="grandRow" type="xsd:boolean" default="false"/>
        <xsd:attribute name="grandCol" type="xsd:boolean" default="false"/>
        <xsd:attribute name="cacheIndex" type="xsd:boolean" default="false"/>
        <xsd:attribute name="outline" type="xsd:boolean" default="true"/>
        <xsd:attribute name="offset" type="ST_Ref"/>
        <xsd:attribute name="collapsedLevelsAreSubtotals" type="xsd:boolean" default="false"/>
        <xsd:attribute name="axis" type="ST_Axis" use="optional"/>
        <xsd:attribute name="fieldPosition" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PivotAreaType">
        <xsd:restriction base="xsd:string">
```

```
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="normal"/>
            <xsd:enumeration value="data"/>
            <xsd:enumeration value="all"/>
            <xsd:enumeration value="origin"/>
            <xsd:enumeration value="button"/>
            <xsd:enumeration value="topEnd"/>
            <xsd:enumeration value="topRight"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PivotAreaReferences">
        <xsd:sequence>
            <xsd:element name="reference" maxOccurs="unbounded" type="CT_PivotAreaReference"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotAreaReference">
        <xsd:sequence>
            <xsd:element name="x" minOccurs="0" maxOccurs="unbounded" type="CT_Index"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="field" use="optional" type="xsd:unsignedInt"/>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
        <xsd:attribute name="selected" type="xsd:boolean" default="true"/>
        <xsd:attribute name="byPosition" type="xsd:boolean" default="false"/>
        <xsd:attribute name="relative" type="xsd:boolean" default="false"/>
        <xsd:attribute name="defaultSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="sumSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="countASubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="avgSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="maxSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="minSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="productSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="countSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="stdDevSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="stdDevPSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="varSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="varPSubtotal" type="xsd:boolean" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Index">
        <xsd:attribute name="v" use="required" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Axis">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="axisRow"/>
            <xsd:enumeration value="axisCol"/>
            <xsd:enumeration value="axisPage"/>
            <xsd:enumeration value="axisValues"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="queryTable" type="CT_QueryTable"/>
    <xsd:complexType name="CT_QueryTable">
        <xsd:sequence>
```



```
            <xsd:element name="queryTableRefresh" type="CT_QueryTableRefresh" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="headers" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="rowNumbers" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="disableRefresh" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="backgroundRefresh" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="firstBackgroundRefresh" type="xsd:boolean" use="optional"
          default="false"/>
        <xsd:attribute name="refreshOnLoad" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="growShrinkType" type="ST_GrowShrinkType" use="optional"
          default="insertDelete"/>
        <xsd:attribute name="fillFormulas" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="removeDataOnSave" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="disableEdit" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="preserveFormatting" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="adjustColumnWidth" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="intermediate" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="connectionId" type="xsd:unsignedInt" use="required"/>
        <xsd:attributeGroup ref="AG_AutoFormat"/>
    </xsd:complexType>
    <xsd:complexType name="CT_QueryTableRefresh">
        <xsd:sequence>
            <xsd:element name="queryTableFields" type="CT_QueryTableFields" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="queryTableDeletedFields" type="CT_QueryTableDeletedFields"
              minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sortState" minOccurs="0" maxOccurs="1" type="CT_SortState"/>
            <xsd:element name="extLst" minOccurs="0" maxOccurs="1" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="preserveSortFilterLayout" type="xsd:boolean" use="optional"
          default="true"/>
        <xsd:attribute name="fieldIdWrapped" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="headersInLastRefresh" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="minimumVersion" type="xsd:unsignedByte" use="optional" default="0"/>
        <xsd:attribute name="nextId" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="unboundColumnsLeft" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="unboundColumnsRight" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_QueryTableDeletedFields">
        <xsd:sequence>
            <xsd:element name="deletedField" type="CT_DeletedField" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DeletedField">
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_QueryTableFields">
        <xsd:sequence>
```

```
            <xsd:element name="queryTableField" type="CT_QueryTableField" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_QueryTableField">
        <xsd:sequence minOccurs="0">
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="dataBound" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="rowNumbers" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="fillFormulas" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="clipped" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="tableColumnId" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_GrowShrinkType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="insertDelete"/>
            <xsd:enumeration value="insertClear"/>
            <xsd:enumeration value="overwriteClear"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="sst" type="CT_Sst"/>
    <xsd:complexType name="CT_Sst">
        <xsd:sequence>
            <xsd:element name="si" type="CT_Rst" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="uniqueCount" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PhoneticType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="halfwidthKatakana"/>
            <xsd:enumeration value="fullwidthKatakana"/>
            <xsd:enumeration value="Hiragana"/>
            <xsd:enumeration value="noConversion"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PhoneticAlignment">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="noControl"/>
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="distributed"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PhoneticRun">
        <xsd:sequence>
            <xsd:element name="t" type="s:ST_Xstring" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
```



```
        <xsd:attribute name="sb" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="eb" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RElt">
        <xsd:sequence>
            <xsd:element name="rPr" type="CT_RPrElt" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="t" type="s:ST_Xstring" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_RPrElt">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="rFont" type="CT_FontName" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="charset" type="CT_IntProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="family" type="CT_IntProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="b" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="i" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="strike" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="outline" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="shadow" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="condense" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extend" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="color" type="CT_Color" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sz" type="CT_FontSize" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="u" type="CT_UnderlineProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="vertAlign" type="CT_VerticalAlignFontProperty" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="scheme" type="CT_FontScheme" minOccurs="0" maxOccurs="1"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_Rst">
        <xsd:sequence>
            <xsd:element name="t" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="r" type="CT_RElt" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rPh" type="CT_PhoneticRun" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="phoneticPr" minOccurs="0" maxOccurs="1" type="CT_PhoneticPr"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_PhoneticPr">
        <xsd:attribute name="fontId" type="ST_FontId" use="required"/>
        <xsd:attribute name="type" type="ST_PhoneticType" use="optional" default="fullwidthKatakana"/>
        <xsd:attribute name="alignment" type="ST_PhoneticAlignment" use="optional" default="left"/>
    </xsd:complexType>
    <xsd:element name="headers" type="CT_RevisionHeaders"/>
    <xsd:element name="revisions" type="CT_Revisions"/>
    <xsd:complexType name="CT_RevisionHeaders">
        <xsd:sequence>
            <xsd:element name="header" type="CT_RevisionHeader" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="guid" type="s:ST_Guid" use="required"/>
        <xsd:attribute name="lastGuid" type="s:ST_Guid" use="optional"/>
        <xsd:attribute name="shared" type="xsd:boolean" default="true"/>
        <xsd:attribute name="diskRevisions" type="xsd:boolean" default="false"/>
        <xsd:attribute name="history" type="xsd:boolean" default="true"/>
```

```
        <xsd:attribute name="trackRevisions" type="xsd:boolean" default="true"/>
        <xsd:attribute name="exclusive" type="xsd:boolean" default="false"/>
        <xsd:attribute name="revisionId" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="version" type="xsd:int" default="1"/>
        <xsd:attribute name="keepChangeHistory" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="protected" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="preserveHistory" type="xsd:unsignedInt" default="30"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Revisions">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="rrc" type="CT_RevisionRowColumn" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rm" type="CT_RevisionMove" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rcv" type="CT_RevisionCustomView" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rsnm" type="CT_RevisionSheetRename" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="ris" type="CT_RevisionInsertSheet" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="rcc" type="CT_RevisionCellChange" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rfmt" type="CT_RevisionFormatting" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="raf" type="CT_RevisionAutoFormatting" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="rdn" type="CT_RevisionDefinedName" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="rcmt" type="CT_RevisionComment" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rqt" type="CT_RevisionQueryTableField" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="rcft" type="CT_RevisionConflict" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:attributeGroup name="AG_RevData">
        <xsd:attribute name="rId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="ua" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="ra" type="xsd:boolean" use="optional" default="false"/>
    </xsd:attributeGroup>
    <xsd:complexType name="CT_RevisionHeader">
        <xsd:sequence>
            <xsd:element name="sheetIdMap" minOccurs="1" maxOccurs="1" type="CT_SheetIdMap"/>
            <xsd:element name="reviewedList" minOccurs="0" maxOccurs="1" type="CT_ReviewedRevisions"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="guid" type="s:ST_Guid" use="required"/>
        <xsd:attribute name="dateTime" type="xsd:dateTime" use="required"/>
        <xsd:attribute name="maxSheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="userName" type="s:ST_Xstring" use="required"/>
        <xsd:attribute ref="r:id" use="required"/>
        <xsd:attribute name="minRId" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="maxRId" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SheetIdMap">
        <xsd:sequence>
            <xsd:element name="sheetId" type="CT_SheetId" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
```



```
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SheetId">
        <xsd:attribute name="val" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ReviewedRevisions">
        <xsd:sequence>
            <xsd:element name="reviewed" type="CT_Reviewed" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Reviewed">
        <xsd:attribute name="rId" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_UndoInfo">
        <xsd:attribute name="index" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="exp" type="ST_FormulaExpression" use="required"/>
        <xsd:attribute name="ref3D" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="array" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="v" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="nf" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="cs" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="dr" type="ST_RefA" use="required"/>
        <xsd:attribute name="dn" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="r" type="ST_CellRef" use="optional"/>
        <xsd:attribute name="sId" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionRowColumn">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="undo" type="CT_UndoInfo" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rcc" type="CT_RevisionCellChange" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rfmt" type="CT_RevisionFormatting" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:choice>
        <xsd:attributeGroup ref="AG_RevData"/>
        <xsd:attribute name="sId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="eol" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
        <xsd:attribute name="action" type="ST_rwColActionType" use="required"/>
        <xsd:attribute name="edge" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionMove">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="undo" type="CT_UndoInfo" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rcc" type="CT_RevisionCellChange" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rfmt" type="CT_RevisionFormatting" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:choice>
        <xsd:attributeGroup ref="AG_RevData"/>
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="source" type="ST_Ref" use="required"/>
        <xsd:attribute name="destination" type="ST_Ref" use="required"/>
        <xsd:attribute name="sourceSheetId" type="xsd:unsignedInt" use="optional" default="0"/>
```

```
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionCustomView">
        <xsd:attribute name="guid" type="s:ST_Guid" use="required"/>
        <xsd:attribute name="action" type="ST_RevisionAction" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionSheetRename">
        <xsd:sequence>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_RevData"/>
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="oldName" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="newName" type="s:ST_Xstring" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionInsertSheet">
        <xsd:attributeGroup ref="AG_RevData"/>
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="sheetPosition" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionCellChange">
        <xsd:sequence>
            <xsd:element name="oc" type="CT_Cell" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="nc" type="CT_Cell" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="odxf" type="CT_Dxf" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ndxf" type="CT_Dxf" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_RevData"/>
        <xsd:attribute name="sId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="odxf" type="xsd:boolean" default="false"/>
        <xsd:attribute name="xfDxf" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="s" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="dxf" type="xsd:boolean" default="false"/>
        <xsd:attribute name="numFmtId" type="ST_NumFmtId" use="optional"/>
        <xsd:attribute name="quotePrefix" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="oldQuotePrefix" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="ph" type="xsd:boolean" default="false"/>
        <xsd:attribute name="oldPh" type="xsd:boolean" default="false"/>
        <xsd:attribute name="endOfListFormulaUpdate" type="xsd:boolean" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionFormatting">
        <xsd:sequence>
            <xsd:element name="dxf" type="CT_Dxf" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="xfDxf" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="s" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="sqref" type="ST_Sqref" use="required"/>
        <xsd:attribute name="start" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="length" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
```



```
    <xsd:complexType name="CT_RevisionAutoFormatting">
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attributeGroup ref="AG_AutoFormat"/>
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionComment">
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="cell" type="ST_CellRef" use="required"/>
        <xsd:attribute name="guid" type="s:ST_Guid" use="required"/>
        <xsd:attribute name="action" type="ST_RevisionAction" default="add"/>
        <xsd:attribute name="alwaysShow" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="old" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="hiddenRow" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="hiddenColumn" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="author" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="oldLength" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="newLength" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionDefinedName">
        <xsd:sequence>
            <xsd:element name="formula" type="ST_Formula" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="oldFormula" type="ST_Formula" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_RevData"/>
        <xsd:attribute name="localSheetId" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="customView" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="function" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="oldFunction" type="xsd:boolean" default="false"/>
        <xsd:attribute name="functionGroupId" type="xsd:unsignedByte" use="optional"/>
        <xsd:attribute name="oldFunctionGroupId" type="xsd:unsignedByte" use="optional"/>
        <xsd:attribute name="shortcutKey" type="xsd:unsignedByte" use="optional"/>
        <xsd:attribute name="oldShortcutKey" type="xsd:unsignedByte" use="optional"/>
        <xsd:attribute name="hidden" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="oldHidden" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="customMenu" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="oldCustomMenu" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="description" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="oldDescription" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="help" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="oldHelp" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="statusBar" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="oldStatusBar" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="comment" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="oldComment" type="s:ST_Xstring" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionConflict">
        <xsd:attributeGroup ref="AG_RevData"/>
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionQueryTableField">
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="required"/>
```

```
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
        <xsd:attribute name="fieldId" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_rwColActionType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="insertRow"/>
            <xsd:enumeration value="deleteRow"/>
            <xsd:enumeration value="insertCol"/>
            <xsd:enumeration value="deleteCol"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_RevisionAction">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="add"/>
            <xsd:enumeration value="delete"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FormulaExpression">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="ref"/>
            <xsd:enumeration value="refError"/>
            <xsd:enumeration value="area"/>
            <xsd:enumeration value="areaError"/>
            <xsd:enumeration value="computedArea"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="users" type="CT_Users"/>
    <xsd:complexType name="CT_Users">
        <xsd:sequence>
            <xsd:element name="userInfo" minOccurs="0" maxOccurs="256" type="CT_SharedUser"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SharedUser">
        <xsd:sequence>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="guid" type="s:ST_Guid" use="required"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="id" type="xsd:int" use="required"/>
        <xsd:attribute name="dateTime" type="xsd:dateTime" use="required"/>
    </xsd:complexType>
    <xsd:element name="worksheet" type="CT_Worksheet"/>
    <xsd:element name="chartsheet" type="CT_Chartsheet"/>
    <xsd:element name="dialogsheet" type="CT_Dialogsheet"/>
    <xsd:complexType name="CT_Macrosheet">
        <xsd:sequence>
            <xsd:element name="sheetPr" type="CT_SheetPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dimension" type="CT_SheetDimension" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sheetViews" type="CT_SheetViews" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sheetFormatPr" type="CT_SheetFormatPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cols" type="CT_Cols" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="sheetData" type="CT_SheetData" minOccurs="1" maxOccurs="1"/>
```



```
            <xsd:element name="sheetProtection" type="CT_SheetProtection" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="autoFilter" type="CT_AutoFilter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sortState" type="CT_SortState" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dataConsolidate" type="CT_DataConsolidate" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="customSheetViews" type="CT_CustomSheetViews" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="phoneticPr" type="CT_PhoneticPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="conditionalFormatting" type="CT_ConditionalFormatting" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="printOptions" type="CT_PrintOptions" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pageMargins" type="CT_PageMargins" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pageSetup" type="CT_PageSetup" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="headerFooter" type="CT_HeaderFooter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="rowBreaks" type="CT_PageBreak" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="colBreaks" type="CT_PageBreak" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="customProperties" type="CT_CustomProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="drawing" type="CT_Drawing" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="legacyDrawing" type="CT_LegacyDrawing" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="legacyDrawingHF" type="CT_LegacyDrawing" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="drawingHF" type="CT_DrawingHF" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="picture" type="CT_SheetBackgroundPicture" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="oleObjects" type="CT_OleObjects" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Dialogsheet">
        <xsd:sequence>
            <xsd:element name="sheetPr" minOccurs="0" type="CT_SheetPr"/>
            <xsd:element name="sheetViews" minOccurs="0" type="CT_SheetViews"/>
            <xsd:element name="sheetFormatPr" minOccurs="0" type="CT_SheetFormatPr"/>
            <xsd:element name="sheetProtection" type="CT_SheetProtection" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="customSheetViews" minOccurs="0" type="CT_CustomSheetViews"/>
            <xsd:element name="printOptions" minOccurs="0" type="CT_PrintOptions"/>
            <xsd:element name="pageMargins" minOccurs="0" type="CT_PageMargins"/>
            <xsd:element name="pageSetup" minOccurs="0" type="CT_PageSetup"/>
            <xsd:element name="headerFooter" minOccurs="0" type="CT_HeaderFooter"/>
            <xsd:element name="drawing" minOccurs="0" type="CT_Drawing"/>
            <xsd:element name="legacyDrawing" minOccurs="0" type="CT_LegacyDrawing"/>
            <xsd:element name="legacyDrawingHF" type="CT_LegacyDrawing" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="drawingHF" type="CT_DrawingHF" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="oleObjects" type="CT_OleObjects" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="controls" type="CT_Controls" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Worksheet">
        <xsd:sequence>
            <xsd:element name="sheetPr" type="CT_SheetPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dimension" type="CT_SheetDimension" minOccurs="0" maxOccurs="1"/>
```

```
            <xsd:element name="sheetViews" type="CT_SheetViews" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sheetFormatPr" type="CT_SheetFormatPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cols" type="CT_Cols" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="sheetData" type="CT_SheetData" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="sheetCalcPr" type="CT_SheetCalcPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sheetProtection" type="CT_SheetProtection" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="protectedRanges" type="CT_ProtectedRanges" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="scenarios" type="CT_Scenarios" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="autoFilter" type="CT_AutoFilter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sortState" type="CT_SortState" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dataConsolidate" type="CT_DataConsolidate" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="customSheetViews" type="CT_CustomSheetViews" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="mergeCells" type="CT_MergeCells" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="phoneticPr" type="CT_PhoneticPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="conditionalFormatting" type="CT_ConditionalFormatting" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="dataValidations" type="CT_DataValidations" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="hyperlinks" type="CT_Hyperlinks" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="printOptions" type="CT_PrintOptions" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pageMargins" type="CT_PageMargins" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pageSetup" type="CT_PageSetup" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="headerFooter" type="CT_HeaderFooter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="rowBreaks" type="CT_PageBreak" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="colBreaks" type="CT_PageBreak" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="customProperties" type="CT_CustomProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="cellWatches" type="CT_CellWatches" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ignoredErrors" type="CT_IgnoredErrors" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="smartTags" type="CT_SmartTags" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="drawing" type="CT_Drawing" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="legacyDrawing" type="CT_LegacyDrawing" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="legacyDrawingHF" type="CT_LegacyDrawing" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="drawingHF" type="CT_DrawingHF" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="picture" type="CT_SheetBackgroundPicture" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="oleObjects" type="CT_OleObjects" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="controls" type="CT_Controls" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="webPublishItems" type="CT_WebPublishItems" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="tableParts" type="CT_TableParts" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SheetData">
        <xsd:sequence>
            <xsd:element name="row" type="CT_Row" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SheetCalcPr">
```



```
        <xsd:attribute name="fullCalcOnLoad" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SheetFormatPr">
        <xsd:attribute name="baseColWidth" type="xsd:unsignedInt" use="optional" default="8"/>
        <xsd:attribute name="defaultColWidth" type="xsd:double" use="optional"/>
        <xsd:attribute name="defaultRowHeight" type="xsd:double" use="required"/>
        <xsd:attribute name="customHeight" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="zeroHeight" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="thickTop" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="thickBottom" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="outlineLevelRow" type="xsd:unsignedByte" use="optional" default="0"/>
        <xsd:attribute name="outlineLevelCol" type="xsd:unsignedByte" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Cols">
        <xsd:sequence>
            <xsd:element name="col" type="CT_Col" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Col">
        <xsd:attribute name="min" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="max" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="width" type="xsd:double" use="optional"/>
        <xsd:attribute name="style" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="hidden" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="bestFit" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="customWidth" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="phonetic" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="outlineLevel" type="xsd:unsignedByte" use="optional" default="0"/>
        <xsd:attribute name="collapsed" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_CellSpan">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_CellSpans">
        <xsd:list itemType="ST_CellSpan"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_Row">
        <xsd:sequence>
            <xsd:element name="c" type="CT_Cell" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="r" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="spans" type="ST_CellSpans" use="optional"/>
        <xsd:attribute name="s" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="customFormat" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="ht" type="xsd:double" use="optional"/>
        <xsd:attribute name="hidden" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="customHeight" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="outlineLevel" type="xsd:unsignedByte" use="optional" default="0"/>
        <xsd:attribute name="collapsed" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="thickTop" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="thickBot" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="ph" type="xsd:boolean" use="optional" default="false"/>
```

```
    </xsd:complexType>
    <xsd:complexType name="CT_Cell">
        <xsd:sequence>
            <xsd:element name="f" type="CT_CellFormula" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="v" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="is" type="CT_Rst" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="r" type="ST_CellRef" use="optional"/>
        <xsd:attribute name="s" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="t" type="ST_CellType" use="optional" default="n"/>
        <xsd:attribute name="cm" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="vm" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="ph" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_CellType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="b"/>
            <xsd:enumeration value="n"/>
            <xsd:enumeration value="e"/>
            <xsd:enumeration value="s"/>
            <xsd:enumeration value="str"/>
            <xsd:enumeration value="inlineStr"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_CellFormulaType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="normal"/>
            <xsd:enumeration value="array"/>
            <xsd:enumeration value="dataTable"/>
            <xsd:enumeration value="shared"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SheetPr">
        <xsd:sequence>
            <xsd:element name="tabColor" type="CT_Color" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="outlinePr" type="CT_OutlinePr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pageSetUpPr" type="CT_PageSetUpPr" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="syncHorizontal" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="syncVertical" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="syncRef" type="ST_Ref" use="optional"/>
        <xsd:attribute name="transitionEvaluation" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="transitionEntry" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="published" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="codeName" type="xsd:string" use="optional"/>
        <xsd:attribute name="filterMode" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="enableFormatConditionsCalculation" type="xsd:boolean" use="optional"
          default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SheetDimension">
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
    </xsd:complexType>
```



```
    <xsd:complexType name="CT_SheetViews">
        <xsd:sequence>
            <xsd:element name="sheetView" type="CT_SheetView" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SheetView">
        <xsd:sequence>
            <xsd:element name="pane" type="CT_Pane" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="selection" type="CT_Selection" minOccurs="0" maxOccurs="4"/>
            <xsd:element name="pivotSelection" type="CT_PivotSelection" minOccurs="0" maxOccurs="4"/>
            <xsd:element name="extLst" minOccurs="0" maxOccurs="1" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="windowProtection" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showFormulas" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showGridLines" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showRowColHeaders" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showZeros" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="rightToLeft" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="tabSelected" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showRuler" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showOutlineSymbols" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="defaultGridColor" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showWhiteSpace" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="view" type="ST_SheetViewType" use="optional" default="normal"/>
        <xsd:attribute name="topLeftCell" type="ST_CellRef" use="optional"/>
        <xsd:attribute name="colorId" type="xsd:unsignedInt" use="optional" default="64"/>
        <xsd:attribute name="zoomScale" type="xsd:unsignedInt" use="optional" default="100"/>
        <xsd:attribute name="zoomScaleNormal" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="zoomScaleSheetLayoutView" type="xsd:unsignedInt" use="optional"
          default="0"/>
        <xsd:attribute name="zoomScalePageLayoutView" type="xsd:unsignedInt" use="optional"
          default="0"/>
        <xsd:attribute name="workbookViewId" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Pane">
        <xsd:attribute name="xSplit" type="xsd:double" use="optional" default="0"/>
        <xsd:attribute name="ySplit" type="xsd:double" use="optional" default="0"/>
        <xsd:attribute name="topLeftCell" type="ST_CellRef" use="optional"/>
        <xsd:attribute name="activePane" type="ST_Pane" use="optional" default="topLeft"/>
        <xsd:attribute name="state" type="ST_PaneState" use="optional" default="split"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotSelection">
        <xsd:sequence>
            <xsd:element name="pivotArea" type="CT_PivotArea"/>
        </xsd:sequence>
        <xsd:attribute name="pane" type="ST_Pane" use="optional" default="topLeft"/>
        <xsd:attribute name="showHeader" type="xsd:boolean" default="false"/>
        <xsd:attribute name="label" type="xsd:boolean" default="false"/>
        <xsd:attribute name="data" type="xsd:boolean" default="false"/>
        <xsd:attribute name="extendable" type="xsd:boolean" default="false"/>
        <xsd:attribute name="count" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="axis" type="ST_Axis" use="optional"/>
```

```
        <xsd:attribute name="dimension" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="start" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="min" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="max" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="activeRow" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="activeCol" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="previousRow" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="previousCol" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="click" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute ref="r:id" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Selection">
        <xsd:attribute name="pane" type="ST_Pane" use="optional" default="topLeft"/>
        <xsd:attribute name="activeCell" type="ST_CellRef" use="optional"/>
        <xsd:attribute name="activeCellId" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="sqref" type="ST_Sqref" use="optional" default="A1"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Pane">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="bottomRight"/>
            <xsd:enumeration value="topRight"/>
            <xsd:enumeration value="bottomLeft"/>
            <xsd:enumeration value="topLeft"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PageBreak">
        <xsd:sequence>
            <xsd:element name="brk" type="CT_Break" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="manualBreakCount" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Break">
        <xsd:attribute name="id" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="min" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="max" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="man" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pt" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SheetViewType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="normal"/>
            <xsd:enumeration value="pageBreakPreview"/>
            <xsd:enumeration value="pageLayout"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_OutlinePr">
        <xsd:attribute name="applyStyles" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="summaryBelow" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="summaryRight" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showOutlineSymbols" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PageSetUpPr">
```



```
        <xsd:attribute name="autoPageBreaks" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="fitToPage" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DataConsolidate">
        <xsd:sequence>
            <xsd:element name="dataRefs" type="CT_DataRefs" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="function" type="ST_DataConsolidateFunction" use="optional"
          default="sum"/>
        <xsd:attribute name="startLabels" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="leftLabels" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="topLabels" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="link" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DataConsolidateFunction">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="average"/>
            <xsd:enumeration value="count"/>
            <xsd:enumeration value="countNums"/>
            <xsd:enumeration value="max"/>
            <xsd:enumeration value="min"/>
            <xsd:enumeration value="product"/>
            <xsd:enumeration value="stdDev"/>
            <xsd:enumeration value="stdDevp"/>
            <xsd:enumeration value="sum"/>
            <xsd:enumeration value="var"/>
            <xsd:enumeration value="varp"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DataRefs">
        <xsd:sequence>
            <xsd:element name="dataRef" type="CT_DataRef" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DataRef">
        <xsd:attribute name="ref" type="ST_Ref" use="optional"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="sheet" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute ref="r:id" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MergeCells">
        <xsd:sequence>
            <xsd:element name="mergeCell" type="CT_MergeCell" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MergeCell">
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SmartTags">
        <xsd:sequence>
```

```
            <xsd:element name="cellSmartTags" type="CT_CellSmartTags" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CellSmartTags">
        <xsd:sequence>
            <xsd:element name="cellSmartTag" type="CT_CellSmartTag" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="r" type="ST_CellRef" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CellSmartTag">
        <xsd:sequence>
            <xsd:element name="cellSmartTagPr" minOccurs="0" maxOccurs="unbounded"
              type="CT_CellSmartTagPr"/>
        </xsd:sequence>
        <xsd:attribute name="type" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="deleted" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="xmlBased" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CellSmartTagPr">
        <xsd:attribute name="key" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="val" type="s:ST_Xstring" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Drawing">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_LegacyDrawing">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DrawingHF">
        <xsd:attribute ref="r:id" use="required"/>
        <xsd:attribute name="lho" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="lhe" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="lhf" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="cho" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="che" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="chf" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="rho" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="rhe" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="rhf" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="lfo" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="lfe" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="lff" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="cfo" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="cfe" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="cff" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="rfo" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="rfe" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="rff" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomSheetViews">
        <xsd:sequence>
```



```
            <xsd:element name="customSheetView" minOccurs="1" maxOccurs="unbounded"
              type="CT_CustomSheetView"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomSheetView">
        <xsd:sequence>
            <xsd:element name="pane" type="CT_Pane" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="selection" type="CT_Selection" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="rowBreaks" type="CT_PageBreak" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="colBreaks" type="CT_PageBreak" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pageMargins" type="CT_PageMargins" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="printOptions" type="CT_PrintOptions" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pageSetup" type="CT_PageSetup" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="headerFooter" type="CT_HeaderFooter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="autoFilter" type="CT_AutoFilter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="guid" type="s:ST_Guid" use="required"/>
        <xsd:attribute name="scale" type="xsd:unsignedInt" default="100"/>
        <xsd:attribute name="colorId" type="xsd:unsignedInt" default="64"/>
        <xsd:attribute name="showPageBreaks" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showFormulas" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showGridLines" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showRowCol" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="outlineSymbols" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="zeroValues" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="fitToPage" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="printArea" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="filter" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showAutoFilter" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="hiddenRows" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="hiddenColumns" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="state" type="ST_SheetState" default="visible"/>
        <xsd:attribute name="filterUnique" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="view" type="ST_SheetViewType" default="normal"/>
        <xsd:attribute name="showRuler" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="topLeftCell" type="ST_CellRef" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DataValidations">
        <xsd:sequence>
            <xsd:element name="dataValidation" type="CT_DataValidation" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="disablePrompts" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="xWindow" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="yWindow" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DataValidation">
        <xsd:sequence>
            <xsd:element name="formula1" type="ST_Formula" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="formula2" type="ST_Formula" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
```

```
        <xsd:attribute name="type" type="ST_DataValidationType" use="optional" default="none"/>
        <xsd:attribute name="errorStyle" type="ST_DataValidationErrorStyle" use="optional"
          default="stop"/>
        <xsd:attribute name="imeMode" type="ST_DataValidationImeMode" use="optional"
          default="noControl"/>
        <xsd:attribute name="operator" type="ST_DataValidationOperator" use="optional"
          default="between"/>
        <xsd:attribute name="allowBlank" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showDropDown" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showInputMessage" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showErrorMessage" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="errorTitle" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="error" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="promptTitle" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="prompt" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="sqref" type="ST_Sqref" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DataValidationType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="whole"/>
            <xsd:enumeration value="decimal"/>
            <xsd:enumeration value="list"/>
            <xsd:enumeration value="date"/>
            <xsd:enumeration value="time"/>
            <xsd:enumeration value="textLength"/>
            <xsd:enumeration value="custom"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_DataValidationOperator">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="between"/>
            <xsd:enumeration value="notBetween"/>
            <xsd:enumeration value="equal"/>
            <xsd:enumeration value="notEqual"/>
            <xsd:enumeration value="lessThan"/>
            <xsd:enumeration value="lessThanOrEqual"/>
            <xsd:enumeration value="greaterThan"/>
            <xsd:enumeration value="greaterThanOrEqual"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_DataValidationErrorStyle">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="stop"/>
            <xsd:enumeration value="warning"/>
            <xsd:enumeration value="information"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_DataValidationImeMode">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="noControl"/>
            <xsd:enumeration value="off"/>
            <xsd:enumeration value="on"/>
```



```
            <xsd:enumeration value="disabled"/>
            <xsd:enumeration value="hiragana"/>
            <xsd:enumeration value="fullKatakana"/>
            <xsd:enumeration value="halfKatakana"/>
            <xsd:enumeration value="fullAlpha"/>
            <xsd:enumeration value="halfAlpha"/>
            <xsd:enumeration value="fullHangul"/>
            <xsd:enumeration value="halfHangul"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_CfType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="expression"/>
            <xsd:enumeration value="cellIs"/>
            <xsd:enumeration value="colorScale"/>
            <xsd:enumeration value="dataBar"/>
            <xsd:enumeration value="iconSet"/>
            <xsd:enumeration value="top10"/>
            <xsd:enumeration value="uniqueValues"/>
            <xsd:enumeration value="duplicateValues"/>
            <xsd:enumeration value="containsText"/>
            <xsd:enumeration value="notContainsText"/>
            <xsd:enumeration value="beginsWith"/>
            <xsd:enumeration value="endsWith"/>
            <xsd:enumeration value="containsBlanks"/>
            <xsd:enumeration value="notContainsBlanks"/>
            <xsd:enumeration value="containsErrors"/>
            <xsd:enumeration value="notContainsErrors"/>
            <xsd:enumeration value="timePeriod"/>
            <xsd:enumeration value="aboveAverage"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TimePeriod">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="today"/>
            <xsd:enumeration value="yesterday"/>
            <xsd:enumeration value="tomorrow"/>
            <xsd:enumeration value="last7Days"/>
            <xsd:enumeration value="thisMonth"/>
            <xsd:enumeration value="lastMonth"/>
            <xsd:enumeration value="nextMonth"/>
            <xsd:enumeration value="thisWeek"/>
            <xsd:enumeration value="lastWeek"/>
            <xsd:enumeration value="nextWeek"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ConditionalFormattingOperator">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="lessThan"/>
            <xsd:enumeration value="lessThanOrEqual"/>
            <xsd:enumeration value="equal"/>
            <xsd:enumeration value="notEqual"/>
            <xsd:enumeration value="greaterThanOrEqual"/>
```

```
            <xsd:enumeration value="greaterThan"/>
            <xsd:enumeration value="between"/>
            <xsd:enumeration value="notBetween"/>
            <xsd:enumeration value="containsText"/>
            <xsd:enumeration value="notContains"/>
            <xsd:enumeration value="beginsWith"/>
            <xsd:enumeration value="endsWith"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_CfvoType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="num"/>
            <xsd:enumeration value="percent"/>
            <xsd:enumeration value="max"/>
            <xsd:enumeration value="min"/>
            <xsd:enumeration value="formula"/>
            <xsd:enumeration value="percentile"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_ConditionalFormatting">
        <xsd:sequence>
            <xsd:element name="cfRule" type="CT_CfRule" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="pivot" type="xsd:boolean" default="false"/>
        <xsd:attribute name="sqref" type="ST_Sqref"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CfRule">
        <xsd:sequence>
            <xsd:element name="formula" type="ST_Formula" minOccurs="0" maxOccurs="3"/>
            <xsd:element name="colorScale" type="CT_ColorScale" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dataBar" type="CT_DataBar" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="iconSet" type="CT_IconSet" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="type" type="ST_CfType"/>
        <xsd:attribute name="dxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="priority" type="xsd:int" use="required"/>
        <xsd:attribute name="stopIfTrue" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="aboveAverage" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="percent" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="bottom" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="operator" type="ST_ConditionalFormattingOperator" use="optional"/>
        <xsd:attribute name="text" type="xsd:string" use="optional"/>
        <xsd:attribute name="timePeriod" type="ST_TimePeriod" use="optional"/>
        <xsd:attribute name="rank" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="stdDev" type="xsd:int" use="optional"/>
        <xsd:attribute name="equalAverage" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Hyperlinks">
        <xsd:sequence>
            <xsd:element name="hyperlink" type="CT_Hyperlink" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
```



```
    </xsd:complexType>
    <xsd:complexType name="CT_Hyperlink">
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
        <xsd:attribute ref="r:id" use="optional"/>
        <xsd:attribute name="location" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="tooltip" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="display" type="s:ST_Xstring" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CellFormula">
        <xsd:simpleContent>
            <xsd:extension base="ST_Formula">
                <xsd:attribute name="t" type="ST_CellFormulaType" use="optional" default="normal"/>
                <xsd:attribute name="aca" type="xsd:boolean" use="optional" default="false"/>
                <xsd:attribute name="ref" type="ST_Ref" use="optional"/>
                <xsd:attribute name="dt2D" type="xsd:boolean" use="optional" default="false"/>
                <xsd:attribute name="dtr" type="xsd:boolean" use="optional" default="false"/>
                <xsd:attribute name="del1" type="xsd:boolean" use="optional" default="false"/>
                <xsd:attribute name="del2" type="xsd:boolean" use="optional" default="false"/>
                <xsd:attribute name="r1" type="ST_CellRef" use="optional"/>
                <xsd:attribute name="r2" type="ST_CellRef" use="optional"/>
                <xsd:attribute name="ca" type="xsd:boolean" use="optional" default="false"/>
                <xsd:attribute name="si" type="xsd:unsignedInt" use="optional"/>
                <xsd:attribute name="bx" type="xsd:boolean" use="optional" default="false"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:complexType name="CT_ColorScale">
        <xsd:sequence>
            <xsd:element name="cfvo" type="CT_Cfvo" minOccurs="2" maxOccurs="unbounded"/>
            <xsd:element name="color" type="CT_Color" minOccurs="2" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DataBar">
        <xsd:sequence>
            <xsd:element name="cfvo" type="CT_Cfvo" minOccurs="2" maxOccurs="2"/>
            <xsd:element name="color" type="CT_Color" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="minLength" type="xsd:unsignedInt" use="optional" default="10"/>
        <xsd:attribute name="maxLength" type="xsd:unsignedInt" use="optional" default="90"/>
        <xsd:attribute name="showValue" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_IconSet">
        <xsd:sequence>
            <xsd:element name="cfvo" type="CT_Cfvo" minOccurs="2" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="iconSet" type="ST_IconSetType" use="optional" default="3TrafficLights1"/>
        <xsd:attribute name="showValue" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="percent" type="xsd:boolean" default="true"/>
        <xsd:attribute name="reverse" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Cfvo">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
```

```
        </xsd:sequence>
        <xsd:attribute name="type" type="ST_CfvoType" use="required"/>
        <xsd:attribute name="val" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="gte" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PageMargins">
        <xsd:attribute name="left" type="xsd:double" use="required"/>
        <xsd:attribute name="right" type="xsd:double" use="required"/>
        <xsd:attribute name="top" type="xsd:double" use="required"/>
        <xsd:attribute name="bottom" type="xsd:double" use="required"/>
        <xsd:attribute name="header" type="xsd:double" use="required"/>
        <xsd:attribute name="footer" type="xsd:double" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PrintOptions">
        <xsd:attribute name="horizontalCentered" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="verticalCentered" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="headings" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="gridLines" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="gridLinesSet" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PageSetup">
        <xsd:attribute name="paperSize" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="paperHeight" type="s:ST_PositiveUniversalMeasure" use="optional"/>
        <xsd:attribute name="paperWidth" type="s:ST_PositiveUniversalMeasure" use="optional"/>
        <xsd:attribute name="scale" type="xsd:unsignedInt" use="optional" default="100"/>
        <xsd:attribute name="firstPageNumber" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="fitToWidth" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="fitToHeight" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="pageOrder" type="ST_PageOrder" use="optional" default="downThenOver"/>
        <xsd:attribute name="orientation" type="ST_Orientation" use="optional" default="default"/>
        <xsd:attribute name="usePrinterDefaults" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="blackAndWhite" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="draft" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="cellComments" type="ST_CellComments" use="optional" default="none"/>
        <xsd:attribute name="useFirstPageNumber" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="errors" type="ST_PrintError" use="optional" default="displayed"/>
        <xsd:attribute name="horizontalDpi" type="xsd:unsignedInt" use="optional" default="600"/>
        <xsd:attribute name="verticalDpi" type="xsd:unsignedInt" use="optional" default="600"/>
        <xsd:attribute name="copies" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute ref="r:id" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PageOrder">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="downThenOver"/>
            <xsd:enumeration value="overThenDown"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Orientation">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="default"/>
            <xsd:enumeration value="portrait"/>
            <xsd:enumeration value="landscape"/>
        </xsd:restriction>
```



```
    </xsd:simpleType>
    <xsd:simpleType name="ST_CellComments">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="asDisplayed"/>
            <xsd:enumeration value="atEnd"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_HeaderFooter">
        <xsd:sequence>
            <xsd:element name="oddHeader" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="oddFooter" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="evenHeader" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="evenFooter" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="firstHeader" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="firstFooter" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="differentOddEven" type="xsd:boolean" default="false"/>
        <xsd:attribute name="differentFirst" type="xsd:boolean" default="false"/>
        <xsd:attribute name="scaleWithDoc" type="xsd:boolean" default="true"/>
        <xsd:attribute name="alignWithMargins" type="xsd:boolean" default="true"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PrintError">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="displayed"/>
            <xsd:enumeration value="blank"/>
            <xsd:enumeration value="dash"/>
            <xsd:enumeration value="NA"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Scenarios">
        <xsd:sequence>
            <xsd:element name="scenario" type="CT_Scenario" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="current" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="show" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="sqref" type="ST_Sqref" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SheetProtection">
        <xsd:attribute name="password" type="ST_UnsignedShortHex" use="optional"/>
        <xsd:attribute name="algorithmName" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="hashValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="saltValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="spinCount" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="sheet" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="objects" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="scenarios" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="formatCells" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="formatColumns" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="formatRows" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="insertColumns" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="insertRows" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="insertHyperlinks" type="xsd:boolean" use="optional" default="true"/>
```

```
        <xsd:attribute name="deleteColumns" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="deleteRows" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="selectLockedCells" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="sort" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="autoFilter" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="pivotTables" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="selectUnlockedCells" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ProtectedRanges">
        <xsd:sequence>
            <xsd:element name="protectedRange" type="CT_ProtectedRange" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ProtectedRange">
        <xsd:sequence>
            <xsd:element name="securityDescriptor" type="xsd:string" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="password" type="ST_UnsignedShortHex" use="optional"/>
        <xsd:attribute name="sqref" type="ST_Sqref" use="required"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="securityDescriptor" type="xsd:string" use="optional"/>
        <xsd:attribute name="algorithmName" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="hashValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="saltValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="spinCount" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Scenario">
        <xsd:sequence>
            <xsd:element name="inputCells" type="CT_InputCells" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="locked" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="hidden" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="user" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="comment" type="s:ST_Xstring" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_InputCells">
        <xsd:attribute name="r" type="ST_CellRef" use="required"/>
        <xsd:attribute name="deleted" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="undone" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="val" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="numFmtId" type="ST_NumFmtId" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CellWatches">
        <xsd:sequence>
            <xsd:element name="cellWatch" type="CT_CellWatch" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CellWatch">
        <xsd:attribute name="r" type="ST_CellRef" use="required"/>
```



```
    </xsd:complexType>
    <xsd:complexType name="CT_Chartsheet">
        <xsd:sequence>
            <xsd:element name="sheetPr" type="CT_ChartsheetPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sheetViews" type="CT_ChartsheetViews" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="sheetProtection" type="CT_ChartsheetProtection" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="customSheetViews" type="CT_CustomChartsheetViews" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="pageMargins" minOccurs="0" type="CT_PageMargins"/>
            <xsd:element name="pageSetup" type="CT_CsPageSetup" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="headerFooter" minOccurs="0" type="CT_HeaderFooter"/>
            <xsd:element name="drawing" type="CT_Drawing" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="legacyDrawing" type="CT_LegacyDrawing" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="legacyDrawingHF" type="CT_LegacyDrawing" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="drawingHF" type="CT_DrawingHF" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="picture" type="CT_SheetBackgroundPicture" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="webPublishItems" type="CT_WebPublishItems" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ChartsheetPr">
        <xsd:sequence>
            <xsd:element name="tabColor" type="CT_Color" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="published" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="codeName" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ChartsheetViews">
        <xsd:sequence>
            <xsd:element name="sheetView" type="CT_ChartsheetView" minOccurs="1"
              maxOccurs="unbounded"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ChartsheetView">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="tabSelected" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="zoomScale" type="xsd:unsignedInt" default="100" use="optional"/>
        <xsd:attribute name="workbookViewId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="zoomToFit" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ChartsheetProtection">
        <xsd:attribute name="password" type="ST_UnsignedShortHex" use="optional"/>
        <xsd:attribute name="algorithmName" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="hashValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="saltValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="spinCount" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="content" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="objects" type="xsd:boolean" use="optional" default="false"/>
```

```
    </xsd:complexType>
    <xsd:complexType name="CT_CsPageSetup">
        <xsd:attribute name="paperSize" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="paperHeight" type="s:ST_PositiveUniversalMeasure" use="optional"/>
        <xsd:attribute name="paperWidth" type="s:ST_PositiveUniversalMeasure" use="optional"/>
        <xsd:attribute name="firstPageNumber" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="orientation" type="ST_Orientation" use="optional" default="default"/>
        <xsd:attribute name="usePrinterDefaults" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="blackAndWhite" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="draft" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="useFirstPageNumber" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="horizontalDpi" type="xsd:unsignedInt" use="optional" default="600"/>
        <xsd:attribute name="verticalDpi" type="xsd:unsignedInt" use="optional" default="600"/>
        <xsd:attribute name="copies" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute ref="r:id" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomChartsheetViews">
        <xsd:sequence>
            <xsd:element name="customSheetView" minOccurs="0" maxOccurs="unbounded"
              type="CT_CustomChartsheetView"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomChartsheetView">
        <xsd:sequence>
            <xsd:element name="pageMargins" type="CT_PageMargins" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pageSetup" type="CT_CsPageSetup" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="headerFooter" type="CT_HeaderFooter" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="guid" type="s:ST_Guid" use="required"/>
        <xsd:attribute name="scale" type="xsd:unsignedInt" default="100"/>
        <xsd:attribute name="state" type="ST_SheetState" default="visible"/>
        <xsd:attribute name="zoomToFit" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomProperties">
        <xsd:sequence>
            <xsd:element name="customPr" type="CT_CustomProperty" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomProperty">
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OleObjects">
        <xsd:sequence>
            <xsd:element name="oleObject" type="CT_OleObject" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_OleObject">
        <xsd:sequence>
            <xsd:element name="objectPr" type="CT_ObjectPr" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="progId" type="xsd:string" use="optional"/>
```



```
        <xsd:attribute name="dvAspect" type="ST_DvAspect" use="optional" default="DVASPECT_CONTENT"/>
        <xsd:attribute name="link" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="oleUpdate" type="ST_OleUpdate" use="optional"/>
        <xsd:attribute name="autoLoad" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="shapeId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute ref="r:id" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ObjectPr">
        <xsd:sequence>
            <xsd:element name="anchor" type="CT_ObjectAnchor" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="locked" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="defaultSize" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="print" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="disabled" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="uiObject" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="autoFill" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="autoLine" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="autoPict" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="macro" type="ST_Formula" use="optional"/>
        <xsd:attribute name="altText" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="dde" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute ref="r:id" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DvAspect">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="DVASPECT_CONTENT"/>
            <xsd:enumeration value="DVASPECT_ICON"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_OleUpdate">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="OLEUPDATE_ALWAYS"/>
            <xsd:enumeration value="OLEUPDATE_ONCALL"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_WebPublishItems">
        <xsd:sequence>
            <xsd:element name="webPublishItem" type="CT_WebPublishItem" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WebPublishItem">
        <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="divId" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="sourceType" type="ST_WebSourceType" use="required"/>
        <xsd:attribute name="sourceRef" type="ST_Ref" use="optional"/>
        <xsd:attribute name="sourceObject" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="destinationFile" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="title" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="autoRepublish" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
```

```
    <xsd:complexType name="CT_Controls">
        <xsd:sequence>
            <xsd:element name="control" type="CT_Control" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Control">
        <xsd:sequence>
            <xsd:element name="controlPr" type="CT_ControlPr" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="shapeId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute ref="r:id" use="required"/>
        <xsd:attribute name="name" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ControlPr">
        <xsd:sequence>
            <xsd:element name="anchor" type="CT_ObjectAnchor" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="locked" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="defaultSize" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="print" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="disabled" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="recalcAlways" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="uiObject" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="autoFill" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="autoLine" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="autoPict" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="macro" type="ST_Formula" use="optional"/>
        <xsd:attribute name="altText" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="linkedCell" type="ST_Formula" use="optional"/>
        <xsd:attribute name="listFillRange" type="ST_Formula" use="optional"/>
        <xsd:attribute name="cf" type="s:ST_Xstring" use="optional" default="pict"/>
        <xsd:attribute ref="r:id" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_WebSourceType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="sheet"/>
            <xsd:enumeration value="printArea"/>
            <xsd:enumeration value="autoFilter"/>
            <xsd:enumeration value="range"/>
            <xsd:enumeration value="chart"/>
            <xsd:enumeration value="pivotTable"/>
            <xsd:enumeration value="query"/>
            <xsd:enumeration value="label"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_IgnoredErrors">
        <xsd:sequence>
            <xsd:element name="ignoredError" type="CT_IgnoredError" minOccurs="1"
              maxOccurs="unbounded"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_IgnoredError">
```



```
        <xsd:attribute name="sqref" type="ST_Sqref" use="required"/>
        <xsd:attribute name="evalError" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="twoDigitTextYear" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="numberStoredAsText" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="formula" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="formulaRange" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="unlockedFormula" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="emptyCellReference" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="listDataValidation" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="calculatedColumn" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PaneState">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="split"/>
            <xsd:enumeration value="frozen"/>
            <xsd:enumeration value="frozenSplit"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TableParts">
        <xsd:sequence>
            <xsd:element name="tablePart" type="CT_TablePart" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TablePart">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:element name="metadata" type="CT_Metadata"/>
    <xsd:complexType name="CT_Metadata">
        <xsd:sequence>
            <xsd:element name="metadataTypes" type="CT_MetadataTypes" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="metadataStrings" type="CT_MetadataStrings" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="mdxMetadata" type="CT_MdxMetadata" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="futureMetadata" type="CT_FutureMetadata" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="cellMetadata" type="CT_MetadataBlocks" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="valueMetadata" type="CT_MetadataBlocks" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" maxOccurs="1" type="CT_ExtensionList"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_MetadataTypes">
        <xsd:sequence>
            <xsd:element name="metadataType" type="CT_MetadataType" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MetadataType">
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="minSupportedVersion" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="ghostRow" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="ghostCol" type="xsd:boolean" use="optional" default="false"/>
```

```
        <xsd:attribute name="edit" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="delete" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="copy" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pasteAll" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pasteFormulas" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pasteValues" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pasteFormats" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pasteComments" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pasteDataValidation" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pasteBorders" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pasteColWidths" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pasteNumberFormats" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="merge" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="splitFirst" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="splitAll" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="rowColShift" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="clearAll" type="xsd:boolean" default="false"/>
        <xsd:attribute name="clearFormats" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="clearContents" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="clearComments" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="assign" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="coerce" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="adjust" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="cellMeta" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MetadataBlocks">
        <xsd:sequence>
            <xsd:element name="bk" type="CT_MetadataBlock" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MetadataBlock">
        <xsd:sequence>
            <xsd:element name="rc" type="CT_MetadataRecord" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_MetadataRecord">
        <xsd:attribute name="t" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="v" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FutureMetadata">
        <xsd:sequence>
            <xsd:element name="bk" type="CT_FutureMetadataBlock" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="extLst" minOccurs="0" maxOccurs="1" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FutureMetadataBlock">
        <xsd:sequence>
            <xsd:element name="extLst" minOccurs="0" maxOccurs="1" type="CT_ExtensionList"/>
        </xsd:sequence>
    </xsd:complexType>
```



```
    <xsd:complexType name="CT_MdxMetadata">
        <xsd:sequence>
            <xsd:element name="mdx" type="CT_Mdx" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Mdx">
        <xsd:choice minOccurs="1" maxOccurs="1">
            <xsd:element name="t" type="CT_MdxTuple"/>
            <xsd:element name="ms" type="CT_MdxSet"/>
            <xsd:element name="p" type="CT_MdxMemeberProp"/>
            <xsd:element name="k" type="CT_MdxKPI"/>
        </xsd:choice>
        <xsd:attribute name="n" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="f" type="ST_MdxFunctionType" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_MdxFunctionType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="m"/>
            <xsd:enumeration value="v"/>
            <xsd:enumeration value="s"/>
            <xsd:enumeration value="c"/>
            <xsd:enumeration value="r"/>
            <xsd:enumeration value="p"/>
            <xsd:enumeration value="k"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_MdxTuple">
        <xsd:sequence>
            <xsd:element name="n" type="CT_MetadataStringIndex" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="c" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="ct" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="si" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="fi" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="bc" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="fc" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="i" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="u" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="st" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="b" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MdxSet">
        <xsd:sequence>
            <xsd:element name="n" type="CT_MetadataStringIndex" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="ns" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="c" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="o" type="ST_MdxSetOrder" use="optional" default="u"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_MdxSetOrder">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="u"/>
```

```
            <xsd:enumeration value="a"/>
            <xsd:enumeration value="d"/>
            <xsd:enumeration value="aa"/>
            <xsd:enumeration value="ad"/>
            <xsd:enumeration value="na"/>
            <xsd:enumeration value="nd"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_MdxMemeberProp">
        <xsd:attribute name="n" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="np" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MdxKPI">
        <xsd:attribute name="n" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="np" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="p" type="ST_MdxKPIProperty" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_MdxKPIProperty">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="v"/>
            <xsd:enumeration value="g"/>
            <xsd:enumeration value="s"/>
            <xsd:enumeration value="t"/>
            <xsd:enumeration value="w"/>
            <xsd:enumeration value="m"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_MetadataStringIndex">
        <xsd:attribute name="x" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="s" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MetadataStrings">
        <xsd:sequence>
            <xsd:element name="s" type="CT_XStringElement" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:element name="singleXmlCells" type="CT_SingleXmlCells"/>
    <xsd:complexType name="CT_SingleXmlCells">
        <xsd:sequence>
            <xsd:element name="singleXmlCell" type="CT_SingleXmlCell" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SingleXmlCell">
        <xsd:sequence>
            <xsd:element name="xmlCellPr" type="CT_XmlCellPr" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="r" type="ST_CellRef" use="required"/>
        <xsd:attribute name="connectionId" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_XmlCellPr">
```



```
        <xsd:sequence>
            <xsd:element name="xmlPr" type="CT_XmlPr" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="uniqueName" type="s:ST_Xstring" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_XmlPr">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="mapId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="xpath" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="xmlDataType" type="ST_XmlDataType" use="required"/>
    </xsd:complexType>
    <xsd:element name="styleSheet" type="CT_Stylesheet"/>
    <xsd:complexType name="CT_Stylesheet">
        <xsd:sequence>
            <xsd:element name="numFmts" type="CT_NumFmts" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="fonts" type="CT_Fonts" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="fills" type="CT_Fills" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="borders" type="CT_Borders" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cellStyleXfs" type="CT_CellStyleXfs" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cellXfs" type="CT_CellXfs" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cellStyles" type="CT_CellStyles" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dxfs" type="CT_Dxfs" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tableStyles" type="CT_TableStyles" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="colors" type="CT_Colors" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CellAlignment">
        <xsd:attribute name="horizontal" type="ST_HorizontalAlignment" use="optional"/>
        <xsd:attribute name="vertical" type="ST_VerticalAlignment" use="optional"/>
        <xsd:attribute name="textRotation" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="wrapText" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="indent" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="relativeIndent" type="xsd:int" use="optional"/>
        <xsd:attribute name="justifyLastLine" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="shrinkToFit" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="readingOrder" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_BorderStyle">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="thin"/>
            <xsd:enumeration value="medium"/>
            <xsd:enumeration value="dashed"/>
            <xsd:enumeration value="dotted"/>
            <xsd:enumeration value="thick"/>
            <xsd:enumeration value="double"/>
            <xsd:enumeration value="hair"/>
            <xsd:enumeration value="mediumDashed"/>
```

```
            <xsd:enumeration value="dashDot"/>
            <xsd:enumeration value="mediumDashDot"/>
            <xsd:enumeration value="dashDotDot"/>
            <xsd:enumeration value="mediumDashDotDot"/>
            <xsd:enumeration value="slantDashDot"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Borders">
        <xsd:sequence>
            <xsd:element name="border" type="CT_Border" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Border">
        <xsd:sequence>
            <xsd:element name="start" type="CT_BorderPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="end" type="CT_BorderPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="left" type="CT_BorderPr" minOccurs="0"/>
            <xsd:element name="right" type="CT_BorderPr" minOccurs="0"/>
            <xsd:element name="top" type="CT_BorderPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bottom" type="CT_BorderPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="diagonal" type="CT_BorderPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="vertical" type="CT_BorderPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="horizontal" type="CT_BorderPr" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="diagonalUp" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="diagonalDown" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="outline" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_BorderPr">
        <xsd:sequence>
            <xsd:element name="color" type="CT_Color" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="style" type="ST_BorderStyle" use="optional" default="none"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CellProtection">
        <xsd:attribute name="locked" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="hidden" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Fonts">
        <xsd:sequence>
            <xsd:element name="font" type="CT_Font" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Fills">
        <xsd:sequence>
            <xsd:element name="fill" type="CT_Fill" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Fill">
        <xsd:choice minOccurs="1" maxOccurs="1">
```



```
            <xsd:element name="patternFill" type="CT_PatternFill" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="gradientFill" type="CT_GradientFill" minOccurs="0" maxOccurs="1"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_PatternFill">
        <xsd:sequence>
            <xsd:element name="fgColor" type="CT_Color" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bgColor" type="CT_Color" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="patternType" type="ST_PatternType" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Color">
        <xsd:attribute name="auto" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="indexed" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="rgb" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="theme" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="tint" type="xsd:double" use="optional" default="0.0"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PatternType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="solid"/>
            <xsd:enumeration value="mediumGray"/>
            <xsd:enumeration value="darkGray"/>
            <xsd:enumeration value="lightGray"/>
            <xsd:enumeration value="darkHorizontal"/>
            <xsd:enumeration value="darkVertical"/>
            <xsd:enumeration value="darkDown"/>
            <xsd:enumeration value="darkUp"/>
            <xsd:enumeration value="darkGrid"/>
            <xsd:enumeration value="darkTrellis"/>
            <xsd:enumeration value="lightHorizontal"/>
            <xsd:enumeration value="lightVertical"/>
            <xsd:enumeration value="lightDown"/>
            <xsd:enumeration value="lightUp"/>
            <xsd:enumeration value="lightGrid"/>
            <xsd:enumeration value="lightTrellis"/>
            <xsd:enumeration value="gray125"/>
            <xsd:enumeration value="gray0625"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_GradientFill">
        <xsd:sequence>
            <xsd:element name="stop" type="CT_GradientStop" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="type" type="ST_GradientType" use="optional" default="linear"/>
        <xsd:attribute name="degree" type="xsd:double" use="optional" default="0"/>
        <xsd:attribute name="left" type="xsd:double" use="optional" default="0"/>
        <xsd:attribute name="right" type="xsd:double" use="optional" default="0"/>
        <xsd:attribute name="top" type="xsd:double" use="optional" default="0"/>
        <xsd:attribute name="bottom" type="xsd:double" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GradientStop">
```

```
        <xsd:sequence>
            <xsd:element name="color" type="CT_Color" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="position" type="xsd:double" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_GradientType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="linear"/>
            <xsd:enumeration value="path"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_HorizontalAlignment">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="general"/>
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="right"/>
            <xsd:enumeration value="fill"/>
            <xsd:enumeration value="justify"/>
            <xsd:enumeration value="centerContinuous"/>
            <xsd:enumeration value="distributed"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_VerticalAlignment">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="top"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="bottom"/>
            <xsd:enumeration value="justify"/>
            <xsd:enumeration value="distributed"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_NumFmts">
        <xsd:sequence>
            <xsd:element name="numFmt" type="CT_NumFmt" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NumFmt">
        <xsd:attribute name="numFmtId" type="ST_NumFmtId" use="required"/>
        <xsd:attribute name="formatCode" type="s:ST_Xstring" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CellStyleXfs">
        <xsd:sequence>
            <xsd:element name="xf" type="CT_Xf" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CellXfs">
        <xsd:sequence>
            <xsd:element name="xf" type="CT_Xf" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
```



```
    </xsd:complexType>
    <xsd:complexType name="CT_Xf">
        <xsd:sequence>
            <xsd:element name="alignment" type="CT_CellAlignment" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="protection" type="CT_CellProtection" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="numFmtId" type="ST_NumFmtId" use="optional"/>
        <xsd:attribute name="fontId" type="ST_FontId" use="optional"/>
        <xsd:attribute name="fillId" type="ST_FillId" use="optional"/>
        <xsd:attribute name="borderId" type="ST_BorderId" use="optional"/>
        <xsd:attribute name="xfId" type="ST_CellStyleXfId" use="optional"/>
        <xsd:attribute name="quotePrefix" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pivotButton" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="applyNumberFormat" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="applyFont" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="applyFill" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="applyBorder" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="applyAlignment" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="applyProtection" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CellStyles">
        <xsd:sequence>
            <xsd:element name="cellStyle" type="CT_CellStyle" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CellStyle">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="xfId" type="ST_CellStyleXfId" use="required"/>
        <xsd:attribute name="builtinId" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="iLevel" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="hidden" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="customBuiltin" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Dxfs">
        <xsd:sequence>
            <xsd:element name="dxf" type="CT_Dxf" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Dxf">
        <xsd:sequence>
            <xsd:element name="font" type="CT_Font" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="numFmt" type="CT_NumFmt" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="fill" type="CT_Fill" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="alignment" type="CT_CellAlignment" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="border" type="CT_Border" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="protection" type="CT_CellProtection" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
```

```
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_NumFmtId">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FontId">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FillId">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_BorderId">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_CellStyleXfId">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_DxfId">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_Colors">
        <xsd:sequence>
            <xsd:element name="indexedColors" type="CT_IndexedColors" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="mruColors" type="CT_MRUColors" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_IndexedColors">
        <xsd:sequence>
            <xsd:element name="rgbColor" type="CT_RgbColor" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_MRUColors">
        <xsd:sequence>
            <xsd:element name="color" type="CT_Color" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_RgbColor">
        <xsd:attribute name="rgb" type="ST_UnsignedIntHex" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TableStyles">
        <xsd:sequence>
            <xsd:element name="tableStyle" type="CT_TableStyle" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="defaultTableStyle" type="xsd:string" use="optional"/>
        <xsd:attribute name="defaultPivotStyle" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TableStyle">
        <xsd:sequence>
            <xsd:element name="tableStyleElement" type="CT_TableStyleElement" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="required"/>
```



```
        <xsd:attribute name="pivot" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="table" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TableStyleElement">
        <xsd:attribute name="type" type="ST_TableStyleType" use="required"/>
        <xsd:attribute name="size" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="dxfId" type="ST_DxfId" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TableStyleType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="wholeTable"/>
            <xsd:enumeration value="headerRow"/>
            <xsd:enumeration value="totalRow"/>
            <xsd:enumeration value="firstColumn"/>
            <xsd:enumeration value="lastColumn"/>
            <xsd:enumeration value="firstRowStripe"/>
            <xsd:enumeration value="secondRowStripe"/>
            <xsd:enumeration value="firstColumnStripe"/>
            <xsd:enumeration value="secondColumnStripe"/>
            <xsd:enumeration value="firstHeaderCell"/>
            <xsd:enumeration value="lastHeaderCell"/>
            <xsd:enumeration value="firstTotalCell"/>
            <xsd:enumeration value="lastTotalCell"/>
            <xsd:enumeration value="firstSubtotalColumn"/>
            <xsd:enumeration value="secondSubtotalColumn"/>
            <xsd:enumeration value="thirdSubtotalColumn"/>
            <xsd:enumeration value="firstSubtotalRow"/>
            <xsd:enumeration value="secondSubtotalRow"/>
            <xsd:enumeration value="thirdSubtotalRow"/>
            <xsd:enumeration value="blankRow"/>
            <xsd:enumeration value="firstColumnSubheading"/>
            <xsd:enumeration value="secondColumnSubheading"/>
            <xsd:enumeration value="thirdColumnSubheading"/>
            <xsd:enumeration value="firstRowSubheading"/>
            <xsd:enumeration value="secondRowSubheading"/>
            <xsd:enumeration value="thirdRowSubheading"/>
            <xsd:enumeration value="pageFieldLabels"/>
            <xsd:enumeration value="pageFieldValues"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_BooleanProperty">
        <xsd:attribute name="val" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FontSize">
        <xsd:attribute name="val" type="xsd:double" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_IntProperty">
        <xsd:attribute name="val" type="xsd:int" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FontName">
        <xsd:attribute name="val" type="s:ST_Xstring" use="required"/>
    </xsd:complexType>
```

```
    <xsd:complexType name="CT_VerticalAlignFontProperty">
        <xsd:attribute name="val" type="s:ST_VerticalAlignRun" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FontScheme">
        <xsd:attribute name="val" type="ST_FontScheme" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_FontScheme">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="major"/>
            <xsd:enumeration value="minor"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_UnderlineProperty">
        <xsd:attribute name="val" type="ST_UnderlineValues" use="optional" default="single"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_UnderlineValues">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="single"/>
            <xsd:enumeration value="double"/>
            <xsd:enumeration value="singleAccounting"/>
            <xsd:enumeration value="doubleAccounting"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Font">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="name" type="CT_FontName" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="charset" type="CT_IntProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="family" type="CT_FontFamily" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="b" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="i" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="strike" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="outline" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="shadow" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="condense" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extend" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="color" type="CT_Color" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sz" type="CT_FontSize" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="u" type="CT_UnderlineProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="vertAlign" type="CT_VerticalAlignFontProperty" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="scheme" type="CT_FontScheme" minOccurs="0" maxOccurs="1"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_FontFamily">
        <xsd:attribute name="val" type="ST_FontFamily" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_FontFamily">
        <xsd:restriction base="xsd:integer">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="14"/>
        </xsd:restriction>
```



```
    </xsd:simpleType>
    <xsd:attributeGroup name="AG_AutoFormat">
        <xsd:attribute name="autoFormatId" type="xsd:unsignedInt"/>
        <xsd:attribute name="applyNumberFormats" type="xsd:boolean"/>
        <xsd:attribute name="applyBorderFormats" type="xsd:boolean"/>
        <xsd:attribute name="applyFontFormats" type="xsd:boolean"/>
        <xsd:attribute name="applyPatternFormats" type="xsd:boolean"/>
        <xsd:attribute name="applyAlignmentFormats" type="xsd:boolean"/>
        <xsd:attribute name="applyWidthHeightFormats" type="xsd:boolean"/>
    </xsd:attributeGroup>
    <xsd:element name="externalLink" type="CT_ExternalLink"/>
    <xsd:complexType name="CT_ExternalLink">
      <xsd:sequence>
        <xsd:choice>
            <xsd:element name="externalBook" type="CT_ExternalBook" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ddeLink" type="CT_DdeLink" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="oleLink" type="CT_OleLink" minOccurs="0" maxOccurs="1"/>
        </xsd:choice>
        <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalBook">
        <xsd:sequence>
            <xsd:element name="sheetNames" type="CT_ExternalSheetNames" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="definedNames" type="CT_ExternalDefinedNames" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="sheetDataSet" type="CT_ExternalSheetDataSet" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalSheetNames">
        <xsd:sequence>
            <xsd:element name="sheetName" minOccurs="1" maxOccurs="unbounded"
              type="CT_ExternalSheetName"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalSheetName">
        <xsd:attribute name="val" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalDefinedNames">
        <xsd:sequence>
            <xsd:element name="definedName" type="CT_ExternalDefinedName" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalDefinedName">
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="refersTo" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalSheetDataSet">
        <xsd:sequence>
```

```
            <xsd:element name="sheetData" type="CT_ExternalSheetData" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalSheetData">
        <xsd:sequence>
            <xsd:element name="row" type="CT_ExternalRow" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="refreshError" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalRow">
        <xsd:sequence>
            <xsd:element name="cell" type="CT_ExternalCell" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="r" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalCell">
        <xsd:sequence>
            <xsd:element name="v" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="r" type="ST_CellRef" use="optional"/>
        <xsd:attribute name="t" type="ST_CellType" use="optional" default="n"/>
        <xsd:attribute name="vm" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DdeLink">
        <xsd:sequence>
            <xsd:element name="ddeItems" type="CT_DdeItems" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="ddeService" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="ddeTopic" type="s:ST_Xstring" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DdeItems">
        <xsd:sequence>
            <xsd:element name="ddeItem" type="CT_DdeItem" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DdeItem">
        <xsd:sequence>
            <xsd:element name="values" type="CT_DdeValues" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="s:ST_Xstring" default="0"/>
        <xsd:attribute name="ole" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="advise" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="preferPic" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DdeValues">
        <xsd:sequence>
            <xsd:element name="value" minOccurs="1" maxOccurs="unbounded" type="CT_DdeValue"/>
        </xsd:sequence>
        <xsd:attribute name="rows" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="cols" type="xsd:unsignedInt" use="optional" default="1"/>
    </xsd:complexType>
```



```
    <xsd:complexType name="CT_DdeValue">
        <xsd:sequence>
            <xsd:element name="val" type="s:ST_Xstring" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="t" type="ST_DdeValueType" use="optional" default="n"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DdeValueType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="nil"/>
            <xsd:enumeration value="b"/>
            <xsd:enumeration value="n"/>
            <xsd:enumeration value="e"/>
            <xsd:enumeration value="str"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_OleLink">
        <xsd:sequence>
            <xsd:element name="oleItems" type="CT_OleItems" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute ref="r:id" use="required"/>
        <xsd:attribute name="progId" type="s:ST_Xstring" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OleItems">
        <xsd:sequence>
            <xsd:element name="oleItem" type="CT_OleItem" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_OleItem">
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="icon" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="advise" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="preferPic" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:element name="table" type="CT_Table"/>
    <xsd:complexType name="CT_Table">
        <xsd:sequence>
            <xsd:element name="autoFilter" type="CT_AutoFilter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sortState" type="CT_SortState" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tableColumns" type="CT_TableColumns" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="tableStyleInfo" type="CT_TableStyleInfo" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="displayName" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="comment" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
        <xsd:attribute name="tableType" type="ST_TableType" use="optional" default="worksheet"/>
        <xsd:attribute name="headerRowCount" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="insertRow" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="insertRowShift" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="totalsRowCount" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="totalsRowShown" type="xsd:boolean" use="optional" default="true"/>
```

```
        <xsd:attribute name="published" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="headerRowDxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="dataDxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="totalsRowDxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="headerRowBorderDxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="tableBorderDxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="totalsRowBorderDxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="headerRowCellStyle" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="dataCellStyle" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="totalsRowCellStyle" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="connectionId" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TableType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="worksheet"/>
            <xsd:enumeration value="xml"/>
            <xsd:enumeration value="queryTable"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TableStyleInfo">
        <xsd:attribute name="name" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="showFirstColumn" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="showLastColumn" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="showRowStripes" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="showColumnStripes" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TableColumns">
        <xsd:sequence>
            <xsd:element name="tableColumn" type="CT_TableColumn" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TableColumn">
        <xsd:sequence>
            <xsd:element name="calculatedColumnFormula" type="CT_TableFormula" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="totalsRowFormula" type="CT_TableFormula" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="xmlColumnPr" type="CT_XmlColumnPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="uniqueName" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="totalsRowFunction" type="ST_TotalsRowFunction" use="optional"
          default="none"/>
        <xsd:attribute name="totalsRowLabel" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="queryTableFieldId" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="headerRowDxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="dataDxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="totalsRowDxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="headerRowCellStyle" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="dataCellStyle" type="s:ST_Xstring" use="optional"/>
```



```
        <xsd:attribute name="totalsRowCellStyle" type="s:ST_Xstring" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TableFormula">
        <xsd:simpleContent>
            <xsd:extension base="ST_Formula">
                <xsd:attribute name="array" type="xsd:boolean" default="false"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:simpleType name="ST_TotalsRowFunction">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="sum"/>
            <xsd:enumeration value="min"/>
            <xsd:enumeration value="max"/>
            <xsd:enumeration value="average"/>
            <xsd:enumeration value="count"/>
            <xsd:enumeration value="countNums"/>
            <xsd:enumeration value="stdDev"/>
            <xsd:enumeration value="var"/>
            <xsd:enumeration value="custom"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_XmlColumnPr">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="mapId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="xpath" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="denormalized" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="xmlDataType" type="ST_XmlDataType" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_XmlDataType">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:element name="volTypes" type="CT_VolTypes"/>
    <xsd:complexType name="CT_VolTypes">
        <xsd:sequence>
            <xsd:element name="volType" type="CT_VolType" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_VolType">
        <xsd:sequence>
            <xsd:element name="main" type="CT_VolMain" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="type" type="ST_VolDepType" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_VolMain">
        <xsd:sequence>
            <xsd:element name="tp" type="CT_VolTopic" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="first" type="s:ST_Xstring" use="required"/>
```

```
    </xsd:complexType>
    <xsd:complexType name="CT_VolTopic">
        <xsd:sequence>
            <xsd:element name="v" type="s:ST_Xstring" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="stp" type="s:ST_Xstring" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="tr" type="CT_VolTopicRef" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="t" type="ST_VolValueType" use="optional" default="n"/>
    </xsd:complexType>
    <xsd:complexType name="CT_VolTopicRef">
        <xsd:attribute name="r" type="ST_CellRef" use="required"/>
        <xsd:attribute name="s" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_VolDepType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="realTimeData"/>
            <xsd:enumeration value="olapFunctions"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_VolValueType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="b"/>
            <xsd:enumeration value="n"/>
            <xsd:enumeration value="e"/>
            <xsd:enumeration value="s"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="workbook" type="CT_Workbook"/>
    <xsd:complexType name="CT_Workbook">
        <xsd:sequence>
            <xsd:element name="fileVersion" type="CT_FileVersion" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="fileSharing" type="CT_FileSharing" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="workbookPr" type="CT_WorkbookPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="workbookProtection" type="CT_WorkbookProtection" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="bookViews" type="CT_BookViews" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sheets" type="CT_Sheets" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="functionGroups" type="CT_FunctionGroups" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="externalReferences" type="CT_ExternalReferences" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="definedNames" type="CT_DefinedNames" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="calcPr" type="CT_CalcPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="oleSize" type="CT_OleSize" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="customWorkbookViews" type="CT_CustomWorkbookViews" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="pivotCaches" type="CT_PivotCaches" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="smartTagPr" type="CT_SmartTagPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="smartTagTypes" type="CT_SmartTagTypes" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="webPublishing" type="CT_WebPublishing" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="fileRecoveryPr" type="CT_FileRecoveryPr" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="webPublishObjects" type="CT_WebPublishObjects" minOccurs="0"
              maxOccurs="1"/>
```



```
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="conformance" type="s:ST_ConformanceClass"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FileVersion">
        <xsd:attribute name="appName" type="xsd:string" use="optional"/>
        <xsd:attribute name="lastEdited" type="xsd:string" use="optional"/>
        <xsd:attribute name="lowestEdited" type="xsd:string" use="optional"/>
        <xsd:attribute name="rupBuild" type="xsd:string" use="optional"/>
        <xsd:attribute name="codeName" type="s:ST_Guid" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_BookViews">
        <xsd:sequence>
            <xsd:element name="workbookView" type="CT_BookView" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_BookView">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="visibility" type="ST_Visibility" use="optional" default="visible"/>
        <xsd:attribute name="minimized" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showHorizontalScroll" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showVerticalScroll" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showSheetTabs" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="xWindow" type="xsd:int" use="optional"/>
        <xsd:attribute name="yWindow" type="xsd:int" use="optional"/>
        <xsd:attribute name="windowWidth" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="windowHeight" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="tabRatio" type="xsd:unsignedInt" use="optional" default="600"/>
        <xsd:attribute name="firstSheet" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="activeTab" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="autoFilterDateGrouping" type="xsd:boolean" use="optional"
          default="true"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Visibility">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="visible"/>
            <xsd:enumeration value="hidden"/>
            <xsd:enumeration value="veryHidden"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_CustomWorkbookViews">
        <xsd:sequence>
            <xsd:element name="customWorkbookView" minOccurs="1" maxOccurs="unbounded"
              type="CT_CustomWorkbookView"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomWorkbookView">
        <xsd:sequence>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
```

```
        <xsd:attribute name="guid" type="s:ST_Guid" use="required"/>
        <xsd:attribute name="autoUpdate" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="mergeInterval" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="changesSavedWin" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="onlySync" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="personalView" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="includePrintSettings" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="includeHiddenRowCol" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="maximized" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="minimized" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showHorizontalScroll" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showVerticalScroll" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showSheetTabs" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="xWindow" type="xsd:int" use="optional" default="0"/>
        <xsd:attribute name="yWindow" type="xsd:int" use="optional" default="0"/>
        <xsd:attribute name="windowWidth" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="windowHeight" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="tabRatio" type="xsd:unsignedInt" use="optional" default="600"/>
        <xsd:attribute name="activeSheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="showFormulaBar" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showStatusbar" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showComments" type="ST_Comments" use="optional" default="commIndicator"/>
        <xsd:attribute name="showObjects" type="ST_Objects" use="optional" default="all"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Comments">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="commNone"/>
            <xsd:enumeration value="commIndicator"/>
            <xsd:enumeration value="commIndAndComment"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Objects">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="all"/>
            <xsd:enumeration value="placeholders"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Sheets">
        <xsd:sequence>
            <xsd:element name="sheet" type="CT_Sheet" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Sheet">
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="state" type="ST_SheetState" use="optional" default="visible"/>
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SheetState">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="visible"/>
            <xsd:enumeration value="hidden"/>
```



```
            <xsd:enumeration value="veryHidden"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_WorkbookPr">
        <xsd:attribute name="date1904" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showObjects" type="ST_Objects" use="optional" default="all"/>
        <xsd:attribute name="showBorderUnselectedTables" type="xsd:boolean" use="optional"
          default="true"/>
        <xsd:attribute name="filterPrivacy" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="promptedSolutions" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showInkAnnotation" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="backupFile" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="saveExternalLinkValues" type="xsd:boolean" use="optional"
          default="true"/>
        <xsd:attribute name="updateLinks" type="ST_UpdateLinks" use="optional" default="userSet"/>
        <xsd:attribute name="codeName" type="xsd:string" use="optional"/>
        <xsd:attribute name="hidePivotFieldList" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showPivotChartFilter" type="xsd:boolean" default="false"/>
        <xsd:attribute name="allowRefreshQuery" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="publishItems" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="checkCompatibility" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="autoCompressPictures" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="refreshAllConnections" type="xsd:boolean" use="optional"
          default="false"/>
        <xsd:attribute name="defaultThemeVersion" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_UpdateLinks">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="userSet"/>
            <xsd:enumeration value="never"/>
            <xsd:enumeration value="always"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SmartTagPr">
        <xsd:attribute name="embed" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="show" type="ST_SmartTagShow" use="optional" default="all"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SmartTagShow">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="all"/>
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="noIndicator"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SmartTagTypes">
        <xsd:sequence>
            <xsd:element name="smartTagType" type="CT_SmartTagType" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SmartTagType">
        <xsd:attribute name="namespaceUri" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="optional"/>
```

```
        <xsd:attribute name="url" type="s:ST_Xstring" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FileRecoveryPr">
        <xsd:attribute name="autoRecover" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="crashSave" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="dataExtractLoad" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="repairLoad" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CalcPr">
        <xsd:attribute name="calcId" type="xsd:unsignedInt"/>
        <xsd:attribute name="calcMode" type="ST_CalcMode" use="optional" default="auto"/>
        <xsd:attribute name="fullCalcOnLoad" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="refMode" type="ST_RefMode" use="optional" default="A1"/>
        <xsd:attribute name="iterate" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="iterateCount" type="xsd:unsignedInt" use="optional" default="100"/>
        <xsd:attribute name="iterateDelta" type="xsd:double" use="optional" default="0.001"/>
        <xsd:attribute name="fullPrecision" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="calcCompleted" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="calcOnSave" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="concurrentCalc" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="concurrentManualCount" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="forceFullCalc" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_CalcMode">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="manual"/>
            <xsd:enumeration value="auto"/>
            <xsd:enumeration value="autoNoTable"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_RefMode">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="A1"/>
            <xsd:enumeration value="R1C1"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DefinedNames">
        <xsd:sequence>
            <xsd:element name="definedName" type="CT_DefinedName" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DefinedName">
        <xsd:simpleContent>
            <xsd:extension base="ST_Formula">
                <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
                <xsd:attribute name="comment" type="s:ST_Xstring" use="optional"/>
                <xsd:attribute name="customMenu" type="s:ST_Xstring" use="optional"/>
                <xsd:attribute name="description" type="s:ST_Xstring" use="optional"/>
                <xsd:attribute name="help" type="s:ST_Xstring" use="optional"/>
                <xsd:attribute name="statusBar" type="s:ST_Xstring" use="optional"/>
                <xsd:attribute name="localSheetId" type="xsd:unsignedInt" use="optional"/>
                <xsd:attribute name="hidden" type="xsd:boolean" use="optional" default="false"/>
```



```
                <xsd:attribute name="function" type="xsd:boolean" use="optional" default="false"/>
                <xsd:attribute name="vbProcedure" type="xsd:boolean" use="optional" default="false"/>
                <xsd:attribute name="xlm" type="xsd:boolean" use="optional" default="false"/>
                <xsd:attribute name="functionGroupId" type="xsd:unsignedInt" use="optional"/>
                <xsd:attribute name="shortcutKey" type="s:ST_Xstring" use="optional"/>
                <xsd:attribute name="publishToServer" type="xsd:boolean" use="optional"
                  default="false"/>
                <xsd:attribute name="workbookParameter" type="xsd:boolean" use="optional"
                  default="false"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalReferences">
        <xsd:sequence>
            <xsd:element name="externalReference" type="CT_ExternalReference" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalReference">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SheetBackgroundPicture">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotCaches">
        <xsd:sequence>
            <xsd:element name="pivotCache" type="CT_PivotCache" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotCache">
        <xsd:attribute name="cacheId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FileSharing">
        <xsd:attribute name="readOnlyRecommended" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="userName" type="s:ST_Xstring"/>
        <xsd:attribute name="reservationPassword" type="ST_UnsignedShortHex"/>
        <xsd:attribute name="algorithmName" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="hashValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="saltValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="spinCount" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OleSize">
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WorkbookProtection">
        <xsd:attribute name="workbookPassword" type="ST_UnsignedShortHex" use="optional"/>
        <xsd:attribute name="workbookPasswordCharacterSet" type="xsd:string" use="optional"/>
        <xsd:attribute name="revisionsPassword" type="ST_UnsignedShortHex" use="optional"/>
        <xsd:attribute name="revisionsPasswordCharacterSet" type="xsd:string" use="optional"/>
        <xsd:attribute name="lockStructure" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="lockWindows" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="lockRevision" type="xsd:boolean" use="optional" default="false"/>
```

```
        <xsd:attribute name="revisionsAlgorithmName" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="revisionsHashValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="revisionsSaltValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="revisionsSpinCount" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="workbookAlgorithmName" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="workbookHashValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="workbookSaltValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="workbookSpinCount" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WebPublishing">
        <xsd:attribute name="css" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="thicket" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="longFileNames" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="vml" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="allowPng" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="targetScreenSize" type="ST_TargetScreenSize" use="optional"
          default="800x600"/>
        <xsd:attribute name="dpi" type="xsd:unsignedInt" use="optional" default="96"/>
        <xsd:attribute name="codePage" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="characterSet" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TargetScreenSize">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="544x376"/>
            <xsd:enumeration value="640x480"/>
            <xsd:enumeration value="720x512"/>
            <xsd:enumeration value="800x600"/>
            <xsd:enumeration value="1024x768"/>
            <xsd:enumeration value="1152x882"/>
            <xsd:enumeration value="1152x900"/>
            <xsd:enumeration value="1280x1024"/>
            <xsd:enumeration value="1600x1200"/>
            <xsd:enumeration value="1800x1440"/>
            <xsd:enumeration value="1920x1200"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_FunctionGroups">
        <xsd:sequence maxOccurs="unbounded">
            <xsd:element name="functionGroup" type="CT_FunctionGroup" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="builtInGroupCount" type="xsd:unsignedInt" default="16" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FunctionGroup">
        <xsd:attribute name="name" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WebPublishObjects">
        <xsd:sequence>
            <xsd:element name="webPublishObject" type="CT_WebPublishObject" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WebPublishObject">
```



```
        <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="divId" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="sourceObject" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="destinationFile" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="title" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="autoRepublish" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
</xsd:schema>
```

A.3 PresentationML

This schema is available in the file pml.xsd.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.openxmlformats.org/presentationml/2006/main"
  xmlns:p="http://schemas.openxmlformats.org/presentationml/2006/main"
  xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
  xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
  xmlns:s="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
  elementFormDefault="qualified"
  targetNamespace="http://schemas.openxmlformats.org/presentationml/2006/main">
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
      schemaLocation="shared-relationshipReference.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/drawingml/2006/main" schemaLocation="dml-
      main.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
    <xsd:simpleType name="ST_TransitionSideDirectionType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="l"/>
            <xsd:enumeration value="u"/>
            <xsd:enumeration value="r"/>
            <xsd:enumeration value="d"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TransitionCornerDirectionType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="lu"/>
            <xsd:enumeration value="ru"/>
            <xsd:enumeration value="ld"/>
            <xsd:enumeration value="rd"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TransitionInOutDirectionType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="out"/>
            <xsd:enumeration value="in"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SideDirectionTransition">
        <xsd:attribute name="dir" type="ST_TransitionSideDirectionType" use="optional" default="l"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CornerDirectionTransition">
```

```
        <xsd:attribute name="dir" type="ST_TransitionCornerDirectionType" use="optional"
          default="lu"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TransitionEightDirectionType">
        <xsd:union memberTypes="ST_TransitionSideDirectionType ST_TransitionCornerDirectionType"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_EightDirectionTransition">
        <xsd:attribute name="dir" type="ST_TransitionEightDirectionType" use="optional" default="l"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OrientationTransition">
        <xsd:attribute name="dir" type="ST_Direction" use="optional" default="horz"/>
    </xsd:complexType>
    <xsd:complexType name="CT_InOutTransition">
        <xsd:attribute name="dir" type="ST_TransitionInOutDirectionType" use="optional"
          default="out"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OptionalBlackTransition">
        <xsd:attribute name="thruBlk" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SplitTransition">
        <xsd:attribute name="orient" type="ST_Direction" use="optional" default="horz"/>
        <xsd:attribute name="dir" type="ST_TransitionInOutDirectionType" use="optional"
          default="out"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WheelTransition">
        <xsd:attribute name="spokes" type="xsd:unsignedInt" use="optional" default="4"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TransitionStartSoundAction">
        <xsd:sequence>
            <xsd:element minOccurs="1" maxOccurs="1" name="snd" type="a:CT_EmbeddedWAVAudioFile"/>
        </xsd:sequence>
        <xsd:attribute name="loop" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TransitionSoundAction">
        <xsd:choice minOccurs="1" maxOccurs="1">
            <xsd:element name="stSnd" type="CT_TransitionStartSoundAction"/>
            <xsd:element name="endSnd" type="CT_Empty"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:simpleType name="ST_TransitionSpeed">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="slow"/>
            <xsd:enumeration value="med"/>
            <xsd:enumeration value="fast"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SlideTransition">
        <xsd:sequence>
            <xsd:choice minOccurs="0" maxOccurs="1">
                <xsd:element name="blinds" type="CT_OrientationTransition"/>
                <xsd:element name="checker" type="CT_OrientationTransition"/>
                <xsd:element name="circle" type="CT_Empty"/>
                <xsd:element name="dissolve" type="CT_Empty"/>
```



```
                <xsd:element name="comb" type="CT_OrientationTransition"/>
                <xsd:element name="cover" type="CT_EightDirectionTransition"/>
                <xsd:element name="cut" type="CT_OptionalBlackTransition"/>
                <xsd:element name="diamond" type="CT_Empty"/>
                <xsd:element name="fade" type="CT_OptionalBlackTransition"/>
                <xsd:element name="newsflash" type="CT_Empty"/>
                <xsd:element name="plus" type="CT_Empty"/>
                <xsd:element name="pull" type="CT_EightDirectionTransition"/>
                <xsd:element name="push" type="CT_SideDirectionTransition"/>
                <xsd:element name="random" type="CT_Empty"/>
                <xsd:element name="randomBar" type="CT_OrientationTransition"/>
                <xsd:element name="split" type="CT_SplitTransition"/>
                <xsd:element name="strips" type="CT_CornerDirectionTransition"/>
                <xsd:element name="wedge" type="CT_Empty"/>
                <xsd:element name="wheel" type="CT_WheelTransition"/>
                <xsd:element name="wipe" type="CT_SideDirectionTransition"/>
                <xsd:element name="zoom" type="CT_InOutTransition"/>
            </xsd:choice>
            <xsd:element name="sndAc" minOccurs="0" maxOccurs="1" type="CT_TransitionSoundAction"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="spd" type="ST_TransitionSpeed" use="optional" default="fast"/>
        <xsd:attribute name="advClick" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="advTm" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLTimeIndefinite">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="indefinite"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLTime">
        <xsd:union memberTypes="xsd:unsignedInt ST_TLTimeIndefinite"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLTimeNodeID">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLIterateIntervalTime">
        <xsd:attribute name="val" type="ST_TLTime" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLIterateIntervalPercentage">
        <xsd:attribute name="val" type="a:ST_PositivePercentage" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_IterateType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="el"/>
            <xsd:enumeration value="wd"/>
            <xsd:enumeration value="lt"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLIterateData">
        <xsd:choice minOccurs="1" maxOccurs="1">
            <xsd:element name="tmAbs" type="CT_TLIterateIntervalTime"/>
            <xsd:element name="tmPct" type="CT_TLIterateIntervalPercentage"/>
```

```
        </xsd:choice>
        <xsd:attribute name="type" type="ST_IterateType" use="optional" default="el"/>
        <xsd:attribute name="backwards" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLSubShapeId">
        <xsd:attribute name="spid" type="a:ST_ShapeID" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLTextTargetElement">
        <xsd:choice minOccurs="0" maxOccurs="1">
            <xsd:element name="charRg" type="CT_IndexRange"/>
            <xsd:element name="pRg" type="CT_IndexRange"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLChartSubelementType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="gridLegend"/>
            <xsd:enumeration value="series"/>
            <xsd:enumeration value="category"/>
            <xsd:enumeration value="ptInSeries"/>
            <xsd:enumeration value="ptInCategory"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLOleChartTargetElement">
        <xsd:attribute name="type" type="ST_TLChartSubelementType" use="required"/>
        <xsd:attribute name="lvl" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLShapeTargetElement">
        <xsd:choice minOccurs="0" maxOccurs="1">
            <xsd:element name="bg" type="CT_Empty"/>
            <xsd:element name="subSp" type="CT_TLSubShapeId"/>
            <xsd:element name="oleChartEl" type="CT_TLOleChartTargetElement"/>
            <xsd:element name="txEl" type="CT_TLTextTargetElement"/>
            <xsd:element name="graphicEl" type="a:CT_AnimationElementChoice"/>
        </xsd:choice>
        <xsd:attribute name="spid" type="a:ST_DrawingElementId" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLTimeTargetElement">
        <xsd:choice minOccurs="1" maxOccurs="1">
            <xsd:element name="sldTgt" type="CT_Empty"/>
            <xsd:element name="sndTgt" type="a:CT_EmbeddedWAVAudioFile"/>
            <xsd:element name="spTgt" type="CT_TLShapeTargetElement"/>
            <xsd:element name="inkTgt" type="CT_TLSubShapeId"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_TLTriggerTimeNodeID">
        <xsd:attribute name="val" type="ST_TLTimeNodeID" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLTriggerRuntimeNode">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="first"/>
            <xsd:enumeration value="last"/>
            <xsd:enumeration value="all"/>
        </xsd:restriction>
```



```
    </xsd:simpleType>
    <xsd:complexType name="CT_TLTriggerRuntimeNode">
        <xsd:attribute name="val" type="ST_TLTriggerRuntimeNode" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLTriggerEvent">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="onBegin"/>
            <xsd:enumeration value="onEnd"/>
            <xsd:enumeration value="begin"/>
            <xsd:enumeration value="end"/>
            <xsd:enumeration value="onClick"/>
            <xsd:enumeration value="onDblClick"/>
            <xsd:enumeration value="onMouseOver"/>
            <xsd:enumeration value="onMouseOut"/>
            <xsd:enumeration value="onNext"/>
            <xsd:enumeration value="onPrev"/>
            <xsd:enumeration value="onStopAudio"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLTimeCondition">
        <xsd:choice minOccurs="0" maxOccurs="1">
            <xsd:element name="tgtEl" type="CT_TLTimeTargetElement"/>
            <xsd:element name="tn" type="CT_TLTriggerTimeNodeID"/>
            <xsd:element name="rtn" type="CT_TLTriggerRuntimeNode"/>
        </xsd:choice>
        <xsd:attribute name="evt" use="optional" type="ST_TLTriggerEvent"/>
        <xsd:attribute name="delay" type="ST_TLTime" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLTimeConditionList">
        <xsd:sequence>
            <xsd:element name="cond" type="CT_TLTimeCondition" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TimeNodeList">
        <xsd:choice minOccurs="1" maxOccurs="unbounded">
            <xsd:element name="par" type="CT_TLTimeNodeParallel"/>
            <xsd:element name="seq" type="CT_TLTimeNodeSequence"/>
            <xsd:element name="excl" type="CT_TLTimeNodeExclusive"/>
            <xsd:element name="anim" type="CT_TLAnimateBehavior"/>
            <xsd:element name="animClr" type="CT_TLAnimateColorBehavior"/>
            <xsd:element name="animEffect" type="CT_TLAnimateEffectBehavior"/>
            <xsd:element name="animMotion" type="CT_TLAnimateMotionBehavior"/>
            <xsd:element name="animRot" type="CT_TLAnimateRotationBehavior"/>
            <xsd:element name="animScale" type="CT_TLAnimateScaleBehavior"/>
            <xsd:element name="cmd" type="CT_TLCommandBehavior"/>
            <xsd:element name="set" type="CT_TLSetBehavior"/>
            <xsd:element name="audio" type="CT_TLMediaNodeAudio"/>
            <xsd:element name="video" type="CT_TLMediaNodeVideo"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLTimeNodePresetClassType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="entr"/>
```

```
            <xsd:enumeration value="exit"/>
            <xsd:enumeration value="emph"/>
            <xsd:enumeration value="path"/>
            <xsd:enumeration value="verb"/>
            <xsd:enumeration value="mediacall"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLTimeNodeRestartType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="always"/>
            <xsd:enumeration value="whenNotActive"/>
            <xsd:enumeration value="never"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLTimeNodeFillType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="remove"/>
            <xsd:enumeration value="freeze"/>
            <xsd:enumeration value="hold"/>
            <xsd:enumeration value="transition"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLTimeNodeSyncType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="canSlip"/>
            <xsd:enumeration value="locked"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLTimeNodeMasterRelation">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="sameClick"/>
            <xsd:enumeration value="lastClick"/>
            <xsd:enumeration value="nextClick"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLTimeNodeType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="clickEffect"/>
            <xsd:enumeration value="withEffect"/>
            <xsd:enumeration value="afterEffect"/>
            <xsd:enumeration value="mainSeq"/>
            <xsd:enumeration value="interactiveSeq"/>
            <xsd:enumeration value="clickPar"/>
            <xsd:enumeration value="withGroup"/>
            <xsd:enumeration value="afterGroup"/>
            <xsd:enumeration value="tmRoot"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLCommonTimeNodeData">
        <xsd:sequence>
            <xsd:element name="stCondLst" type="CT_TLTimeConditionList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="endCondLst" type="CT_TLTimeConditionList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="endSync" type="CT_TLTimeCondition" minOccurs="0" maxOccurs="1"/>
```



```
            <xsd:element name="iterate" type="CT_TLIterateData" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="childTnLst" type="CT_TimeNodeList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="subTnLst" type="CT_TimeNodeList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="ST_TLTimeNodeID" use="optional"/>
        <xsd:attribute name="presetID" type="xsd:int" use="optional"/>
        <xsd:attribute name="presetClass" type="ST_TLTimeNodePresetClassType" use="optional"/>
        <xsd:attribute name="presetSubtype" type="xsd:int" use="optional"/>
        <xsd:attribute name="dur" type="ST_TLTime" use="optional"/>
        <xsd:attribute name="repeatCount" type="ST_TLTime" use="optional" default="1000"/>
        <xsd:attribute name="repeatDur" type="ST_TLTime" use="optional"/>
        <xsd:attribute name="spd" type="a:ST_Percentage" use="optional" default="100%"/>
        <xsd:attribute name="accel" type="a:ST_PositiveFixedPercentage" use="optional" default="0%"/>
        <xsd:attribute name="decel" type="a:ST_PositiveFixedPercentage" use="optional" default="0%"/>
        <xsd:attribute name="autoRev" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="restart" type="ST_TLTimeNodeRestartType" use="optional"/>
        <xsd:attribute name="fill" type="ST_TLTimeNodeFillType" use="optional"/>
        <xsd:attribute name="syncBehavior" type="ST_TLTimeNodeSyncType" use="optional"/>
        <xsd:attribute name="tmFilter" type="xsd:string" use="optional"/>
        <xsd:attribute name="evtFilter" type="xsd:string" use="optional"/>
        <xsd:attribute name="display" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="masterRel" type="ST_TLTimeNodeMasterRelation" use="optional"/>
        <xsd:attribute name="bldLvl" type="xsd:int" use="optional"/>
        <xsd:attribute name="grpId" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="afterEffect" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="nodeType" type="ST_TLTimeNodeType" use="optional"/>
        <xsd:attribute name="nodePh" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLTimeNodeParallel">
        <xsd:sequence>
            <xsd:element name="cTn" type="CT_TLCommonTimeNodeData" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLNextActionType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="seek"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLPreviousActionType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="skipTimed"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLTimeNodeSequence">
        <xsd:sequence>
            <xsd:element name="cTn" type="CT_TLCommonTimeNodeData" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="prevCondLst" type="CT_TLTimeConditionList" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="nextCondLst" type="CT_TLTimeConditionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
```

```
359         <xsd:attribute name="concurrent" type="xsd:boolean" use="optional"/>
360         <xsd:attribute name="prevAc" type="ST_TLPreviousActionType" use="optional"/>
361         <xsd:attribute name="nextAc" type="ST_TLNextActionType" use="optional"/>
362     </xsd:complexType>
363     <xsd:complexType name="CT_TLTimeNodeExclusive">
364         <xsd:sequence>
365             <xsd:element name="cTn" type="CT_TLCommonTimeNodeData" minOccurs="1" maxOccurs="1"/>
366         </xsd:sequence>
367     </xsd:complexType>
368     <xsd:complexType name="CT_TLBehaviorAttributeNameList">
369         <xsd:sequence>
370             <xsd:element name="attrName" type="xsd:string" minOccurs="1" maxOccurs="unbounded"/>
371         </xsd:sequence>
372     </xsd:complexType>
373     <xsd:simpleType name="ST_TLBehaviorAdditiveType">
374         <xsd:restriction base="xsd:token">
375             <xsd:enumeration value="base"/>
376             <xsd:enumeration value="sum"/>
377             <xsd:enumeration value="repl"/>
378             <xsd:enumeration value="mult"/>
379             <xsd:enumeration value="none"/>
380         </xsd:restriction>
381     </xsd:simpleType>
382     <xsd:simpleType name="ST_TLBehaviorAccumulateType">
383         <xsd:restriction base="xsd:token">
384             <xsd:enumeration value="none"/>
385             <xsd:enumeration value="always"/>
386         </xsd:restriction>
387     </xsd:simpleType>
388     <xsd:simpleType name="ST_TLBehaviorTransformType">
389         <xsd:restriction base="xsd:token">
390             <xsd:enumeration value="pt"/>
391             <xsd:enumeration value="img"/>
392         </xsd:restriction>
393     </xsd:simpleType>
394     <xsd:simpleType name="ST_TLBehaviorOverrideType">
395         <xsd:restriction base="xsd:token">
396             <xsd:enumeration value="normal"/>
397             <xsd:enumeration value="childStyle"/>
398         </xsd:restriction>
399     </xsd:simpleType>
400     <xsd:complexType name="CT_TLCommonBehaviorData">
401         <xsd:sequence>
402             <xsd:element name="cTn" type="CT_TLCommonTimeNodeData" minOccurs="1" maxOccurs="1"/>
403             <xsd:element name="tgtEl" type="CT_TLTimeTargetElement" minOccurs="1" maxOccurs="1"/>
404             <xsd:element name="attrNameLst" type="CT_TLBehaviorAttributeNameList" minOccurs="0"
405               maxOccurs="1"/>
406         </xsd:sequence>
407         <xsd:attribute name="additive" type="ST_TLBehaviorAdditiveType" use="optional"/>
408         <xsd:attribute name="accumulate" type="ST_TLBehaviorAccumulateType" use="optional"/>
409         <xsd:attribute name="xfrmType" type="ST_TLBehaviorTransformType" use="optional"/>
410         <xsd:attribute name="from" type="xsd:string" use="optional"/>
411         <xsd:attribute name="to" type="xsd:string" use="optional"/>
```



```
        <xsd:attribute name="by" type="xsd:string" use="optional"/>
        <xsd:attribute name="rctx" type="xsd:string" use="optional"/>
        <xsd:attribute name="override" type="ST_TLBehaviorOverrideType" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLAnimVariantBooleanVal">
        <xsd:attribute name="val" type="xsd:boolean" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLAnimVariantIntegerVal">
        <xsd:attribute name="val" type="xsd:int" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLAnimVariantFloatVal">
        <xsd:attribute name="val" type="xsd:float" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLAnimVariantStringVal">
        <xsd:attribute name="val" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLAnimVariant">
        <xsd:choice minOccurs="1" maxOccurs="1">
            <xsd:element name="boolVal" type="CT_TLAnimVariantBooleanVal"/>
            <xsd:element name="intVal" type="CT_TLAnimVariantIntegerVal"/>
            <xsd:element name="fltVal" type="CT_TLAnimVariantFloatVal"/>
            <xsd:element name="strVal" type="CT_TLAnimVariantStringVal"/>
            <xsd:element name="clrVal" type="a:CT_Color"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLTimeAnimateValueTime">
        <xsd:union memberTypes="a:ST_PositiveFixedPercentage ST_TLTimeIndefinite"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLTimeAnimateValue">
        <xsd:sequence>
            <xsd:element name="val" type="CT_TLAnimVariant" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="tm" type="ST_TLTimeAnimateValueTime" use="optional"
          default="indefinite"/>
        <xsd:attribute name="fmla" type="xsd:string" use="optional" default=""/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLTimeAnimateValueList">
        <xsd:sequence>
            <xsd:element name="tav" type="CT_TLTimeAnimateValue" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLAnimateBehaviorCalcMode">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="discrete"/>
            <xsd:enumeration value="lin"/>
            <xsd:enumeration value="fmla"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLAnimateBehaviorValueType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="str"/>
            <xsd:enumeration value="num"/>
            <xsd:enumeration value="clr"/>
```

```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLAnimateBehavior">
        <xsd:sequence>
            <xsd:element name="cBhvr" type="CT_TLCommonBehaviorData" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="tavLst" type="CT_TLTimeAnimateValueList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="by" type="xsd:string" use="optional"/>
        <xsd:attribute name="from" type="xsd:string" use="optional"/>
        <xsd:attribute name="to" type="xsd:string" use="optional"/>
        <xsd:attribute name="calcmode" type="ST_TLAnimateBehaviorCalcMode" use="optional"/>
        <xsd:attribute name="valueType" type="ST_TLAnimateBehaviorValueType" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLByRgbColorTransform">
        <xsd:attribute name="r" type="a:ST_FixedPercentage" use="required"/>
        <xsd:attribute name="g" type="a:ST_FixedPercentage" use="required"/>
        <xsd:attribute name="b" type="a:ST_FixedPercentage" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLByHslColorTransform">
        <xsd:attribute name="h" type="a:ST_Angle" use="required"/>
        <xsd:attribute name="s" type="a:ST_FixedPercentage" use="required"/>
        <xsd:attribute name="l" type="a:ST_FixedPercentage" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLByAnimateColorTransform">
        <xsd:choice minOccurs="1" maxOccurs="1">
            <xsd:element name="rgb" type="CT_TLByRgbColorTransform"/>
            <xsd:element name="hsl" type="CT_TLByHslColorTransform"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLAnimateColorSpace">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="rgb"/>
            <xsd:enumeration value="hsl"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLAnimateColorDirection">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="cw"/>
            <xsd:enumeration value="ccw"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLAnimateColorBehavior">
        <xsd:sequence>
            <xsd:element name="cBhvr" type="CT_TLCommonBehaviorData" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="by" type="CT_TLByAnimateColorTransform" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="from" type="a:CT_Color" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="to" type="a:CT_Color" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="clrSpc" type="ST_TLAnimateColorSpace" use="optional"/>
        <xsd:attribute name="dir" type="ST_TLAnimateColorDirection" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLAnimateEffectTransition">
        <xsd:restriction base="xsd:token">
```



```
            <xsd:enumeration value="in"/>
            <xsd:enumeration value="out"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLAnimateEffectBehavior">
        <xsd:sequence>
            <xsd:element name="cBhvr" type="CT_TLCommonBehaviorData" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="progress" type="CT_TLAnimVariant" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="transition" type="ST_TLAnimateEffectTransition" use="optional"/>
        <xsd:attribute name="filter" type="xsd:string" use="optional"/>
        <xsd:attribute name="prLst" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLAnimateMotionBehaviorOrigin">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="parent"/>
            <xsd:enumeration value="layout"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLAnimateMotionPathEditMode">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="relative"/>
            <xsd:enumeration value="fixed"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLPoint">
        <xsd:attribute name="x" type="a:ST_Percentage" use="required"/>
        <xsd:attribute name="y" type="a:ST_Percentage" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLAnimateMotionBehavior">
        <xsd:sequence>
            <xsd:element name="cBhvr" type="CT_TLCommonBehaviorData" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="by" type="CT_TLPoint" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="from" type="CT_TLPoint" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="to" type="CT_TLPoint" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="rCtr" type="CT_TLPoint" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="origin" type="ST_TLAnimateMotionBehaviorOrigin" use="optional"/>
        <xsd:attribute name="path" type="xsd:string" use="optional"/>
        <xsd:attribute name="pathEditMode" type="ST_TLAnimateMotionPathEditMode" use="optional"/>
        <xsd:attribute name="rAng" type="a:ST_Angle" use="optional"/>
        <xsd:attribute name="ptsTypes" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLAnimateRotationBehavior">
        <xsd:sequence>
            <xsd:element name="cBhvr" type="CT_TLCommonBehaviorData" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="by" type="a:ST_Angle" use="optional"/>
        <xsd:attribute name="from" type="a:ST_Angle" use="optional"/>
        <xsd:attribute name="to" type="a:ST_Angle" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLAnimateScaleBehavior">
```

```
        <xsd:sequence>
            <xsd:element name="cBhvr" type="CT_TLCommonBehaviorData" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="by" type="CT_TLPoint" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="from" type="CT_TLPoint" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="to" type="CT_TLPoint" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="zoomContents" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLCommandType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="evt"/>
            <xsd:enumeration value="call"/>
            <xsd:enumeration value="verb"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLCommandBehavior">
        <xsd:sequence>
            <xsd:element name="cBhvr" type="CT_TLCommonBehaviorData" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute type="ST_TLCommandType" name="type" use="optional"/>
        <xsd:attribute name="cmd" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLSetBehavior">
        <xsd:sequence>
            <xsd:element name="cBhvr" type="CT_TLCommonBehaviorData" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="to" type="CT_TLAnimVariant" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TLCommonMediaNodeData">
        <xsd:sequence>
            <xsd:element name="cTn" type="CT_TLCommonTimeNodeData" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="tgtEl" type="CT_TLTimeTargetElement" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="vol" type="a:ST_PositiveFixedPercentage" default="50%" use="optional"/>
        <xsd:attribute name="mute" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="numSld" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="showWhenStopped" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLMediaNodeAudio">
        <xsd:sequence>
            <xsd:element name="cMediaNode" type="CT_TLCommonMediaNodeData" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="isNarration" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLMediaNodeVideo">
        <xsd:sequence>
            <xsd:element name="cMediaNode" type="CT_TLCommonMediaNodeData" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="fullScrn" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:attributeGroup name="AG_TLBuild">
```



```
        <xsd:attribute name="spid" type="a:ST_DrawingElementId" use="required"/>
        <xsd:attribute name="grpId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="uiExpand" type="xsd:boolean" use="optional" default="false"/>
    </xsd:attributeGroup>
    <xsd:complexType name="CT_TLTemplate">
        <xsd:sequence>
            <xsd:element name="tnLst" type="CT_TimeNodeList" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="lvl" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLTemplateList">
        <xsd:sequence>
            <xsd:element name="tmpl" type="CT_TLTemplate" minOccurs="0" maxOccurs="9"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLParaBuildType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="allAtOnce"/>
            <xsd:enumeration value="p"/>
            <xsd:enumeration value="cust"/>
            <xsd:enumeration value="whole"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLBuildParagraph">
        <xsd:sequence>
            <xsd:element name="tmplLst" type="CT_TLTemplateList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_TLBuild"/>
        <xsd:attribute name="build" type="ST_TLParaBuildType" use="optional" default="whole"/>
        <xsd:attribute name="bldLvl" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="animBg" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="autoUpdateAnimBg" type="xsd:boolean" default="true" use="optional"/>
        <xsd:attribute name="rev" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="advAuto" type="ST_TLTime" use="optional" default="indefinite"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLDiagramBuildType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="whole"/>
            <xsd:enumeration value="depthByNode"/>
            <xsd:enumeration value="depthByBranch"/>
            <xsd:enumeration value="breadthByNode"/>
            <xsd:enumeration value="breadthByLvl"/>
            <xsd:enumeration value="cw"/>
            <xsd:enumeration value="cwIn"/>
            <xsd:enumeration value="cwOut"/>
            <xsd:enumeration value="ccw"/>
            <xsd:enumeration value="ccwIn"/>
            <xsd:enumeration value="ccwOut"/>
            <xsd:enumeration value="inByRing"/>
            <xsd:enumeration value="outByRing"/>
            <xsd:enumeration value="up"/>
            <xsd:enumeration value="down"/>
            <xsd:enumeration value="allAtOnce"/>
```

```
            <xsd:enumeration value="cust"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLBuildDiagram">
        <xsd:attributeGroup ref="AG_TLBuild"/>
        <xsd:attribute name="bld" type="ST_TLDiagramBuildType" use="optional" default="whole"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLOleChartBuildType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="allAtOnce"/>
            <xsd:enumeration value="series"/>
            <xsd:enumeration value="category"/>
            <xsd:enumeration value="seriesEl"/>
            <xsd:enumeration value="categoryEl"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLOleBuildChart">
        <xsd:attributeGroup ref="AG_TLBuild"/>
        <xsd:attribute name="bld" type="ST_TLOleChartBuildType" use="optional" default="allAtOnce"/>
        <xsd:attribute name="animBg" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLGraphicalObjectBuild">
        <xsd:choice minOccurs="1" maxOccurs="1">
            <xsd:element name="bldAsOne" type="CT_Empty"/>
            <xsd:element name="bldSub" type="a:CT_AnimationGraphicalObjectBuildProperties"/>
        </xsd:choice>
        <xsd:attributeGroup ref="AG_TLBuild"/>
    </xsd:complexType>
    <xsd:complexType name="CT_BuildList">
        <xsd:choice minOccurs="1" maxOccurs="unbounded">
            <xsd:element name="bldP" type="CT_TLBuildParagraph"/>
            <xsd:element name="bldDgm" type="CT_TLBuildDiagram"/>
            <xsd:element name="bldOleChart" type="CT_TLOleBuildChart"/>
            <xsd:element name="bldGraphic" type="CT_TLGraphicalObjectBuild"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_SlideTiming">
        <xsd:sequence>
            <xsd:element name="tnLst" type="CT_TimeNodeList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bldLst" type="CT_BuildList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Empty"/>
    <xsd:simpleType name="ST_Name">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Direction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="horz"/>
            <xsd:enumeration value="vert"/>
        </xsd:restriction>
    </xsd:simpleType>
```



```
    <xsd:simpleType name="ST_Index">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_IndexRange">
        <xsd:attribute name="st" type="ST_Index" use="required"/>
        <xsd:attribute name="end" type="ST_Index" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SlideRelationshipListEntry">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SlideRelationshipList">
        <xsd:sequence>
            <xsd:element name="sld" type="CT_SlideRelationshipListEntry" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomShowId">
        <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:group name="EG_SlideListChoice">
        <xsd:choice>
            <xsd:element name="sldAll" type="CT_Empty"/>
            <xsd:element name="sldRg" type="CT_IndexRange"/>
            <xsd:element name="custShow" type="CT_CustomShowId"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_CustomerData">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TagsData">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomerDataList">
        <xsd:sequence minOccurs="0" maxOccurs="1">
            <xsd:element name="custData" type="CT_CustomerData" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="tags" type="CT_TagsData" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Extension">
        <xsd:sequence>
            <xsd:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="uri" type="xsd:token" use="required"/>
    </xsd:complexType>
    <xsd:group name="EG_ExtensionList">
        <xsd:sequence>
            <xsd:element name="ext" type="CT_Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_ExtensionList">
        <xsd:sequence>
            <xsd:group ref="EG_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
```

```
    </xsd:complexType>
    <xsd:complexType name="CT_ExtensionListModify">
        <xsd:sequence>
            <xsd:group ref="EG_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="mod" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CommentAuthor">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="name" type="ST_Name" use="required"/>
        <xsd:attribute name="initials" type="ST_Name" use="required"/>
        <xsd:attribute name="lastIdx" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="clrIdx" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CommentAuthorList">
        <xsd:sequence>
            <xsd:element name="cmAuthor" type="CT_CommentAuthor" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="cmAuthorLst" type="CT_CommentAuthorList"/>
    <xsd:complexType name="CT_Comment">
        <xsd:sequence>
            <xsd:element name="pos" type="a:CT_Point2D" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="text" type="xsd:string" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="authorId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="dt" type="xsd:dateTime" use="optional"/>
        <xsd:attribute name="idx" type="ST_Index" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CommentList">
        <xsd:sequence>
            <xsd:element name="cm" type="CT_Comment" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="cmLst" type="CT_CommentList"/>
    <xsd:attributeGroup name="AG_Ole">
        <xsd:attribute name="spid" type="a:ST_ShapeID" use="optional"/>
        <xsd:attribute name="name" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="showAsIcon" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute ref="r:id" use="optional"/>
        <xsd:attribute name="imgW" type="a:ST_PositiveCoordinate32" use="optional"/>
        <xsd:attribute name="imgH" type="a:ST_PositiveCoordinate32" use="optional"/>
    </xsd:attributeGroup>
    <xsd:simpleType name="ST_OleObjectFollowColorScheme">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="full"/>
            <xsd:enumeration value="textAndBackground"/>
        </xsd:restriction>
```



```
    </xsd:simpleType>
    <xsd:complexType name="CT_OleObjectEmbed">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="followColorScheme" type="ST_OleObjectFollowColorScheme" use="optional"
          default="none"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OleObjectLink">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="updateAutomatic" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OleObject">
        <xsd:sequence>
          <xsd:choice minOccurs="1" maxOccurs="1">
              <xsd:element name="embed" type="CT_OleObjectEmbed"/>
              <xsd:element name="link" type="CT_OleObjectLink"/>
          </xsd:choice>
          <xsd:element name="pic" type="CT_Picture" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_Ole"/>
        <xsd:attribute name="progId" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:element name="oleObj" type="CT_OleObject"/>
    <xsd:complexType name="CT_Control">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pic" type="CT_Picture" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_Ole"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ControlList">
        <xsd:sequence>
            <xsd:element name="control" type="CT_Control" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_SlideId">
        <xsd:restriction base="xsd:unsignedInt">
            <xsd:minInclusive value="256"/>
            <xsd:maxExclusive value="2147483648"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SlideIdListEntry">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="ST_SlideId" use="required"/>
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SlideIdList">
        <xsd:sequence>
```

```
            <xsd:element name="sldId" type="CT_SlideIdListEntry" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_SlideMasterId">
        <xsd:restriction base="xsd:unsignedInt">
            <xsd:minInclusive value="2147483648"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SlideMasterIdListEntry">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="ST_SlideMasterId" use="optional"/>
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SlideMasterIdList">
        <xsd:sequence>
            <xsd:element name="sldMasterId" type="CT_SlideMasterIdListEntry" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_NotesMasterIdListEntry">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NotesMasterIdList">
        <xsd:sequence>
            <xsd:element name="notesMasterId" type="CT_NotesMasterIdListEntry" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_HandoutMasterIdListEntry">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_HandoutMasterIdList">
        <xsd:sequence>
            <xsd:element name="handoutMasterId" type="CT_HandoutMasterIdListEntry" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_EmbeddedFontDataId">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_EmbeddedFontListEntry">
        <xsd:sequence>
            <xsd:element name="font" type="a:CT_TextFont" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="regular" type="CT_EmbeddedFontDataId" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bold" type="CT_EmbeddedFontDataId" minOccurs="0" maxOccurs="1"/>
```



```
            <xsd:element name="italic" type="CT_EmbeddedFontDataId" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="boldItalic" type="CT_EmbeddedFontDataId" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_EmbeddedFontList">
        <xsd:sequence>
            <xsd:element name="embeddedFont" type="CT_EmbeddedFontListEntry" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SmartTags">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomShow">
        <xsd:sequence>
            <xsd:element name="sldLst" type="CT_SlideRelationshipList" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="ST_Name" use="required"/>
        <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomShowList">
        <xsd:sequence>
            <xsd:element name="custShow" type="CT_CustomShow" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_PhotoAlbumLayout">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="fitToSlide"/>
            <xsd:enumeration value="1pic"/>
            <xsd:enumeration value="2pic"/>
            <xsd:enumeration value="4pic"/>
            <xsd:enumeration value="1picTitle"/>
            <xsd:enumeration value="2picTitle"/>
            <xsd:enumeration value="4picTitle"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PhotoAlbumFrameShape">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="frameStyle1"/>
            <xsd:enumeration value="frameStyle2"/>
            <xsd:enumeration value="frameStyle3"/>
            <xsd:enumeration value="frameStyle4"/>
            <xsd:enumeration value="frameStyle5"/>
            <xsd:enumeration value="frameStyle6"/>
            <xsd:enumeration value="frameStyle7"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PhotoAlbum">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="bw" type="xsd:boolean" use="optional" default="false"/>
```

```
        <xsd:attribute name="showCaptions" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="layout" type="ST_PhotoAlbumLayout" use="optional" default="fitToSlide"/>
        <xsd:attribute name="frame" type="ST_PhotoAlbumFrameShape" use="optional"
          default="frameStyle1"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SlideSizeCoordinate">
        <xsd:restriction base="a:ST_PositiveCoordinate32">
            <xsd:minInclusive value="914400"/>
            <xsd:maxInclusive value="51206400"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_SlideSizeType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="screen4x3"/>
            <xsd:enumeration value="letter"/>
            <xsd:enumeration value="A4"/>
            <xsd:enumeration value="35mm"/>
            <xsd:enumeration value="overhead"/>
            <xsd:enumeration value="banner"/>
            <xsd:enumeration value="custom"/>
            <xsd:enumeration value="ledger"/>
            <xsd:enumeration value="A3"/>
            <xsd:enumeration value="B4ISO"/>
            <xsd:enumeration value="B5ISO"/>
            <xsd:enumeration value="B4JIS"/>
            <xsd:enumeration value="B5JIS"/>
            <xsd:enumeration value="hagakiCard"/>
            <xsd:enumeration value="screen16x9"/>
            <xsd:enumeration value="screen16x10"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SlideSize">
        <xsd:attribute name="cx" type="ST_SlideSizeCoordinate" use="required"/>
        <xsd:attribute name="cy" type="ST_SlideSizeCoordinate" use="required"/>
        <xsd:attribute name="type" type="ST_SlideSizeType" use="optional" default="custom"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Kinsoku">
        <xsd:attribute name="lang" type="xsd:string" use="optional"/>
        <xsd:attribute name="invalStChars" type="xsd:string" use="required"/>
        <xsd:attribute name="invalEndChars" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_BookmarkIdSeed">
        <xsd:restriction base="xsd:unsignedInt">
            <xsd:minInclusive value="1"/>
            <xsd:maxExclusive value="2147483648"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_ModifyVerifier">
        <xsd:attribute name="algorithmName" type="xsd:string" use="optional"/>
        <xsd:attribute name="hashValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="saltValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="spinValue" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="cryptProviderType" type="s:ST_CryptProv" use="optional"/>
```



```
        <xsd:attribute name="cryptAlgorithmClass" type="s:ST_AlgClass" use="optional"/>
        <xsd:attribute name="cryptAlgorithmType" type="s:ST_AlgType" use="optional"/>
        <xsd:attribute name="cryptAlgorithmSid" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="spinCount" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="saltData" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="hashData" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="cryptProvider" type="xsd:string" use="optional"/>
        <xsd:attribute name="algIdExt" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="algIdExtSource" type="xsd:string" use="optional"/>
        <xsd:attribute name="cryptProviderTypeExt" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="cryptProviderTypeExtSource" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Presentation">
        <xsd:sequence>
            <xsd:element name="sldMasterIdLst" type="CT_SlideMasterIdList" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="notesMasterIdLst" type="CT_NotesMasterIdList" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="handoutMasterIdLst" type="CT_HandoutMasterIdList" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="sldIdLst" type="CT_SlideIdList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sldSz" type="CT_SlideSize" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="notesSz" type="a:CT_PositiveSize2D" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="smartTags" type="CT_SmartTags" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="embeddedFontLst" type="CT_EmbeddedFontList" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="custShowLst" type="CT_CustomShowList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="photoAlbum" type="CT_PhotoAlbum" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="custDataLst" type="CT_CustomerDataList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="kinsoku" type="CT_Kinsoku" minOccurs="0"/>
            <xsd:element name="defaultTextStyle" type="a:CT_TextListStyle" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="modifyVerifier" type="CT_ModifyVerifier" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="serverZoom" type="a:ST_Percentage" use="optional" default="50%"/>
        <xsd:attribute name="firstSlideNum" type="xsd:int" use="optional" default="1"/>
        <xsd:attribute name="showSpecialPlsOnTitleSld" type="xsd:boolean" use="optional"
          default="true"/>
        <xsd:attribute name="rtl" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="removePersonalInfoOnSave" type="xsd:boolean" use="optional"
          default="false"/>
        <xsd:attribute name="compatMode" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="strictFirstAndLastChars" type="xsd:boolean" use="optional"
          default="true"/>
        <xsd:attribute name="embedTrueTypeFonts" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="saveSubsetFonts" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="autoCompressPictures" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="bookmarkIdSeed" type="ST_BookmarkIdSeed" use="optional" default="1"/>
        <xsd:attribute name="conformance" type="s:ST_ConformanceClass"/>
    </xsd:complexType>
    <xsd:element name="presentation" type="CT_Presentation"/>
    <xsd:complexType name="CT_HtmlPublishProperties">
```

```
        <xsd:sequence>
            <xsd:group ref="EG_SlideListChoice" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="showSpeakerNotes" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="target" type="xsd:string" use="optional"/>
        <xsd:attribute name="title" type="xsd:string" use="optional" default=""/>
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_WebColorType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="browser"/>
            <xsd:enumeration value="presentationText"/>
            <xsd:enumeration value="presentationAccent"/>
            <xsd:enumeration value="whiteTextOnBlack"/>
            <xsd:enumeration value="blackTextOnWhite"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_WebScreenSize">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="544x376"/>
            <xsd:enumeration value="640x480"/>
            <xsd:enumeration value="720x512"/>
            <xsd:enumeration value="800x600"/>
            <xsd:enumeration value="1024x768"/>
            <xsd:enumeration value="1152x882"/>
            <xsd:enumeration value="1152x900"/>
            <xsd:enumeration value="1280x1024"/>
            <xsd:enumeration value="1600x1200"/>
            <xsd:enumeration value="1800x1400"/>
            <xsd:enumeration value="1920x1200"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_WebEncoding">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_WebProperties">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="showAnimation" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="resizeGraphics" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="allowPng" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="relyOnVml" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="organizeInFolders" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="useLongFilenames" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="imgSz" type="ST_WebScreenSize" use="optional" default="800x600"/>
        <xsd:attribute name="encoding" type="ST_WebEncoding" use="optional" default=""/>
        <xsd:attribute name="clr" type="ST_WebColorType" use="optional" default="whiteTextOnBlack"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PrintWhat">
        <xsd:restriction base="xsd:token">
```



```
            <xsd:enumeration value="slides"/>
            <xsd:enumeration value="handouts1"/>
            <xsd:enumeration value="handouts2"/>
            <xsd:enumeration value="handouts3"/>
            <xsd:enumeration value="handouts4"/>
            <xsd:enumeration value="handouts6"/>
            <xsd:enumeration value="handouts9"/>
            <xsd:enumeration value="notes"/>
            <xsd:enumeration value="outline"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PrintColorMode">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="bw"/>
            <xsd:enumeration value="gray"/>
            <xsd:enumeration value="clr"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PrintProperties">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="prnWhat" type="ST_PrintWhat" use="optional" default="slides"/>
        <xsd:attribute name="clrMode" type="ST_PrintColorMode" use="optional" default="clr"/>
        <xsd:attribute name="hiddenSlides" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="scaleToFitPaper" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="frameSlides" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ShowInfoBrowse">
        <xsd:attribute name="showScrollbar" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ShowInfoKiosk">
        <xsd:attribute name="restart" type="xsd:unsignedInt" use="optional" default="300000"/>
    </xsd:complexType>
    <xsd:group name="EG_ShowType">
        <xsd:choice>
            <xsd:element name="present" type="CT_Empty"/>
            <xsd:element name="browse" type="CT_ShowInfoBrowse"/>
            <xsd:element name="kiosk" type="CT_ShowInfoKiosk"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_ShowProperties">
        <xsd:sequence minOccurs="0" maxOccurs="1">
            <xsd:group ref="EG_ShowType" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_SlideListChoice" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="penClr" type="a:CT_Color" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="loop" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showNarration" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showAnimation" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="useTimings" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
```

```
    <xsd:complexType name="CT_PresentationProperties">
        <xsd:sequence>
            <xsd:element name="htmlPubPr" type="CT_HtmlPublishProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="webPr" type="CT_WebProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="prnPr" type="CT_PrintProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="showPr" type="CT_ShowProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="clrMru" type="a:CT_ColorMRU" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="presentationPr" type="CT_PresentationProperties"/>
    <xsd:complexType name="CT_HeaderFooter">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="sldNum" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="hdr" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="ftr" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="dt" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PlaceholderType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="title"/>
            <xsd:enumeration value="body"/>
            <xsd:enumeration value="ctrTitle"/>
            <xsd:enumeration value="subTitle"/>
            <xsd:enumeration value="dt"/>
            <xsd:enumeration value="sldNum"/>
            <xsd:enumeration value="ftr"/>
            <xsd:enumeration value="hdr"/>
            <xsd:enumeration value="obj"/>
            <xsd:enumeration value="chart"/>
            <xsd:enumeration value="tbl"/>
            <xsd:enumeration value="clipArt"/>
            <xsd:enumeration value="dgm"/>
            <xsd:enumeration value="media"/>
            <xsd:enumeration value="sldImg"/>
            <xsd:enumeration value="pic"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PlaceholderSize">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="full"/>
            <xsd:enumeration value="half"/>
            <xsd:enumeration value="quarter"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Placeholder">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="type" type="ST_PlaceholderType" use="optional" default="obj"/>
```



```
        <xsd:attribute name="orient" type="ST_Direction" use="optional" default="horz"/>
        <xsd:attribute name="sz" type="ST_PlaceholderSize" use="optional" default="full"/>
        <xsd:attribute name="idx" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="hasCustomPrompt" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ApplicationNonVisualDrawingProps">
        <xsd:sequence>
            <xsd:element name="ph" type="CT_Placeholder" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="a:EG_Media" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="custDataLst" type="CT_CustomerDataList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="isPhoto" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="userDrawn" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ShapeNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvSpPr" type="a:CT_NonVisualDrawingShapeProps" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="nvPr" type="CT_ApplicationNonVisualDrawingProps" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Shape">
        <xsd:sequence>
            <xsd:element name="nvSpPr" type="CT_ShapeNonVisual" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="style" type="a:CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="txBody" type="a:CT_TextBody" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="useBgFill" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ConnectorNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvCxnSpPr" type="a:CT_NonVisualConnectorProperties" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="nvPr" type="CT_ApplicationNonVisualDrawingProps" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Connector">
        <xsd:sequence>
            <xsd:element name="nvCxnSpPr" type="CT_ConnectorNonVisual" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="style" type="a:CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_PictureNonVisual">
        <xsd:sequence>
```

```
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvPicPr" type="a:CT_NonVisualPictureProperties" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="nvPr" type="CT_ApplicationNonVisualDrawingProps" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Picture">
        <xsd:sequence>
            <xsd:element name="nvPicPr" type="CT_PictureNonVisual" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="blipFill" type="a:CT_BlipFillProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="style" type="a:CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GraphicalObjectFrameNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvGraphicFramePr" type="a:CT_NonVisualGraphicFrameProperties"
              minOccurs="1" maxOccurs="1"/>
            <xsd:element name="nvPr" type="CT_ApplicationNonVisualDrawingProps" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GraphicalObjectFrame">
        <xsd:sequence>
            <xsd:element name="nvGraphicFramePr" type="CT_GraphicalObjectFrameNonVisual" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="xfrm" type="a:CT_Transform2D" minOccurs="1" maxOccurs="1"/>
            <xsd:element ref="a:graphic" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="bwMode" type="a:ST_BlackWhiteMode" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupShapeNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvGrpSpPr" type="a:CT_NonVisualGroupDrawingShapeProps" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="nvPr" type="CT_ApplicationNonVisualDrawingProps" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupShape">
        <xsd:sequence>
            <xsd:element name="nvGrpSpPr" type="CT_GroupShapeNonVisual" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="grpSpPr" type="a:CT_GroupShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="sp" type="CT_Shape"/>
                <xsd:element name="grpSp" type="CT_GroupShape"/>
                <xsd:element name="graphicFrame" type="CT_GraphicalObjectFrame"/>
                <xsd:element name="cxnSp" type="CT_Connector"/>
```



```
                <xsd:element name="pic" type="CT_Picture"/>
                <xsd:element name="contentPart" type="CT_Rel"/>
            </xsd:choice>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Rel">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:group name="EG_TopLevelSlide">
        <xsd:sequence>
            <xsd:element name="clrMap" type="a:CT_ColorMapping" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:group name="EG_ChildSlide">
        <xsd:sequence>
            <xsd:element name="clrMapOvr" type="a:CT_ColorMappingOverride" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:attributeGroup name="AG_ChildSlide">
        <xsd:attribute name="showMasterSp" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showMasterPhAnim" type="xsd:boolean" use="optional" default="true"/>
    </xsd:attributeGroup>
    <xsd:complexType name="CT_BackgroundProperties">
        <xsd:sequence>
            <xsd:group ref="a:EG_FillProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:group ref="a:EG_EffectProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="shadeToTitle" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:group name="EG_Background">
        <xsd:choice>
            <xsd:element name="bgPr" type="CT_BackgroundProperties"/>
            <xsd:element name="bgRef" type="a:CT_StyleMatrixReference"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_Background">
        <xsd:sequence>
            <xsd:group ref="EG_Background"/>
        </xsd:sequence>
        <xsd:attribute name="bwMode" type="a:ST_BlackWhiteMode" use="optional" default="white"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CommonSlideData">
        <xsd:sequence>
            <xsd:element name="bg" type="CT_Background" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="spTree" type="CT_GroupShape" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="custDataLst" type="CT_CustomerDataList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="controls" type="CT_ControlList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="optional" default=""/>
```

```
    </xsd:complexType>
    <xsd:complexType name="CT_Slide">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element name="cSld" type="CT_CommonSlideData" minOccurs="1" maxOccurs="1"/>
            <xsd:group ref="EG_ChildSlide" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="transition" type="CT_SlideTransition" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="timing" type="CT_SlideTiming" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_ChildSlide"/>
        <xsd:attribute name="show" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:element name="sld" type="CT_Slide"/>
    <xsd:simpleType name="ST_SlideLayoutType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="title"/>
            <xsd:enumeration value="tx"/>
            <xsd:enumeration value="twoColTx"/>
            <xsd:enumeration value="tbl"/>
            <xsd:enumeration value="txAndChart"/>
            <xsd:enumeration value="chartAndTx"/>
            <xsd:enumeration value="dgm"/>
            <xsd:enumeration value="chart"/>
            <xsd:enumeration value="txAndClipArt"/>
            <xsd:enumeration value="clipArtAndTx"/>
            <xsd:enumeration value="titleOnly"/>
            <xsd:enumeration value="blank"/>
            <xsd:enumeration value="txAndObj"/>
            <xsd:enumeration value="objAndTx"/>
            <xsd:enumeration value="objOnly"/>
            <xsd:enumeration value="obj"/>
            <xsd:enumeration value="txAndMedia"/>
            <xsd:enumeration value="mediaAndTx"/>
            <xsd:enumeration value="objOverTx"/>
            <xsd:enumeration value="txOverObj"/>
            <xsd:enumeration value="txAndTwoObj"/>
            <xsd:enumeration value="twoObjAndTx"/>
            <xsd:enumeration value="twoObjOverTx"/>
            <xsd:enumeration value="fourObj"/>
            <xsd:enumeration value="vertTx"/>
            <xsd:enumeration value="clipArtAndVertTx"/>
            <xsd:enumeration value="vertTitleAndTx"/>
            <xsd:enumeration value="vertTitleAndTxOverChart"/>
            <xsd:enumeration value="twoObj"/>
            <xsd:enumeration value="objAndTwoObj"/>
            <xsd:enumeration value="twoObjAndObj"/>
            <xsd:enumeration value="cust"/>
            <xsd:enumeration value="secHead"/>
            <xsd:enumeration value="twoTxTwoObj"/>
            <xsd:enumeration value="objTx"/>
            <xsd:enumeration value="picTx"/>
        </xsd:restriction>
    </xsd:simpleType>
```



```
    <xsd:complexType name="CT_SlideLayout">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element name="cSld" type="CT_CommonSlideData" minOccurs="1" maxOccurs="1"/>
            <xsd:group ref="EG_ChildSlide" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="transition" type="CT_SlideTransition" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="timing" type="CT_SlideTiming" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="hf" type="CT_HeaderFooter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_ChildSlide"/>
        <xsd:attribute name="matchingName" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="type" type="ST_SlideLayoutType" use="optional" default="cust"/>
        <xsd:attribute name="preserve" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="userDrawn" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:element name="sldLayout" type="CT_SlideLayout"/>
    <xsd:complexType name="CT_SlideMasterTextStyles">
        <xsd:sequence>
            <xsd:element name="titleStyle" type="a:CT_TextListStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bodyStyle" type="a:CT_TextListStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="otherStyle" type="a:CT_TextListStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_SlideLayoutId">
        <xsd:restriction base="xsd:unsignedInt">
            <xsd:minInclusive value="2147483648"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SlideLayoutIdListEntry">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="ST_SlideLayoutId" use="optional"/>
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SlideLayoutIdList">
        <xsd:sequence>
            <xsd:element name="sldLayoutId" type="CT_SlideLayoutIdListEntry" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SlideMaster">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element name="cSld" type="CT_CommonSlideData" minOccurs="1" maxOccurs="1"/>
            <xsd:group ref="EG_TopLevelSlide" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="sldLayoutIdLst" type="CT_SlideLayoutIdList" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="transition" type="CT_SlideTransition" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="timing" type="CT_SlideTiming" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="hf" type="CT_HeaderFooter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="txStyles" type="CT_SlideMasterTextStyles" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
```

```
        </xsd:sequence>
        <xsd:attribute name="preserve" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:element name="sldMaster" type="CT_SlideMaster"/>
    <xsd:complexType name="CT_HandoutMaster">
        <xsd:sequence>
            <xsd:element name="cSld" type="CT_CommonSlideData" minOccurs="1" maxOccurs="1"/>
            <xsd:group ref="EG_TopLevelSlide" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="hf" type="CT_HeaderFooter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="handoutMaster" type="CT_HandoutMaster"/>
    <xsd:complexType name="CT_NotesMaster">
        <xsd:sequence>
            <xsd:element name="cSld" type="CT_CommonSlideData" minOccurs="1" maxOccurs="1"/>
            <xsd:group ref="EG_TopLevelSlide" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="hf" type="CT_HeaderFooter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="notesStyle" type="a:CT_TextListStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="notesMaster" type="CT_NotesMaster"/>
    <xsd:complexType name="CT_NotesSlide">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element name="cSld" type="CT_CommonSlideData" minOccurs="1" maxOccurs="1"/>
            <xsd:group ref="EG_ChildSlide" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_ChildSlide"/>
    </xsd:complexType>
    <xsd:element name="notes" type="CT_NotesSlide"/>
    <xsd:complexType name="CT_SlideSyncProperties">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="serverSldId" type="xsd:string" use="required"/>
        <xsd:attribute name="serverSldModifiedTime" type="xsd:dateTime" use="required"/>
        <xsd:attribute name="clientInsertedTime" type="xsd:dateTime" use="required"/>
    </xsd:complexType>
    <xsd:element name="sldSyncPr" type="CT_SlideSyncProperties"/>
    <xsd:complexType name="CT_StringTag">
        <xsd:attribute name="name" type="xsd:string" use="required"/>
        <xsd:attribute name="val" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TagList">
        <xsd:sequence>
            <xsd:element name="tag" type="CT_StringTag" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="tagLst" type="CT_TagList"/>
    <xsd:simpleType name="ST_SplitterBarState">
        <xsd:restriction base="xsd:token">
```



```
            <xsd:enumeration value="minimized"/>
            <xsd:enumeration value="restored"/>
            <xsd:enumeration value="maximized"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ViewType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="sldView"/>
            <xsd:enumeration value="sldMasterView"/>
            <xsd:enumeration value="notesView"/>
            <xsd:enumeration value="handoutView"/>
            <xsd:enumeration value="notesMasterView"/>
            <xsd:enumeration value="outlineView"/>
            <xsd:enumeration value="sldSorterView"/>
            <xsd:enumeration value="sldThumbnailView"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_NormalViewPortion">
        <xsd:attribute name="sz" type="a:ST_PositiveFixedPercentage" use="required"/>
        <xsd:attribute name="autoAdjust" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NormalViewProperties">
        <xsd:sequence>
            <xsd:element name="restoredLeft" type="CT_NormalViewPortion" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="restoredTop" type="CT_NormalViewPortion" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="showOutlineIcons" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="snapVertSplitter" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="vertBarState" type="ST_SplitterBarState" use="optional"
          default="restored"/>
        <xsd:attribute name="horzBarState" type="ST_SplitterBarState" use="optional"
          default="restored"/>
        <xsd:attribute name="preferSingleView" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CommonViewProperties">
        <xsd:sequence>
            <xsd:element name="scale" type="a:CT_Scale2D" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="origin" type="a:CT_Point2D" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="varScale" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NotesTextViewProperties">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element name="cViewPr" type="CT_CommonViewProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_OutlineViewSlideEntry">
        <xsd:attribute ref="r:id" use="required"/>
        <xsd:attribute name="collapse" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OutlineViewSlideList">
```

```
        <xsd:sequence>
            <xsd:element name="sld" type="CT_OutlineViewSlideEntry" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_OutlineViewProperties">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element name="cViewPr" type="CT_CommonViewProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="sldLst" type="CT_OutlineViewSlideList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SlideSorterViewProperties">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element name="cViewPr" type="CT_CommonViewProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="showFormatting" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Guide">
        <xsd:attribute name="orient" type="ST_Direction" use="optional" default="vert"/>
        <xsd:attribute name="pos" type="a:ST_Coordinate32" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GuideList">
        <xsd:sequence minOccurs="0" maxOccurs="1">
            <xsd:element name="guide" type="CT_Guide" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CommonSlideViewProperties">
        <xsd:sequence>
            <xsd:element name="cViewPr" type="CT_CommonViewProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="guideLst" type="CT_GuideList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="snapToGrid" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="snapToObjects" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showGuides" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SlideViewProperties">
        <xsd:sequence>
            <xsd:element name="cSldViewPr" type="CT_CommonSlideViewProperties" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_NotesViewProperties">
        <xsd:sequence>
            <xsd:element name="cSldViewPr" type="CT_CommonSlideViewProperties" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ViewProperties">
        <xsd:sequence minOccurs="0" maxOccurs="1">
```



```
            <xsd:element name="normalViewPr" type="CT_NormalViewProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="slideViewPr" type="CT_SlideViewProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="outlineViewPr" type="CT_OutlineViewProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="notesTextViewPr" type="CT_NotesTextViewProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="sorterViewPr" type="CT_SlideSorterViewProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="notesViewPr" type="CT_NotesViewProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="gridSpacing" type="a:CT_PositiveSize2D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="lastView" type="ST_ViewType" use="optional" default="sldView"/>
        <xsd:attribute name="showComments" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:element name="viewPr" type="CT_ViewProperties"/>
</xsd:schema>
```

A.4 DrawingML - Framework

A.4.1 DrawingML - Main

This schema is available in the file dml-main.xsd.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
  xmlns:s="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
  xmlns="http://schemas.openxmlformats.org/drawingml/2006/main"
  targetNamespace="http://schemas.openxmlformats.org/drawingml/2006/main"
  elementFormDefault="qualified">
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
      schemaLocation="shared-relationshipReference.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/drawingml/2006/diagram"
      schemaLocation="dml-diagram.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/drawingml/2006/chart"
      schemaLocation="dml-chart.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/drawingml/2006/picture"
      schemaLocation="dml-picture.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/drawingml/2006/lockedCanvas"
      schemaLocation="dml-lockedCanvas.xsd"/>
    <xsd:complexType name="CT_AudioFile">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute ref="r:link" use="required"/>
        <xsd:attribute name="contentType" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_VideoFile">
```

```
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute ref="r:link" use="required"/>
        <xsd:attribute name="contentType" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_QuickTimeFile">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute ref="r:link" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AudioCDTime">
        <xsd:attribute name="track" type="xsd:unsignedByte" use="required"/>
        <xsd:attribute name="time" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AudioCD">
        <xsd:sequence>
            <xsd:element name="st" type="CT_AudioCDTime" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="end" type="CT_AudioCDTime" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_Media">
        <xsd:choice>
            <xsd:element name="audioCd" type="CT_AudioCD"/>
            <xsd:element name="wavAudioFile" type="CT_EmbeddedWAVAudioFile"/>
            <xsd:element name="audioFile" type="CT_AudioFile"/>
            <xsd:element name="videoFile" type="CT_VideoFile"/>
            <xsd:element name="quickTimeFile" type="CT_QuickTimeFile"/>
        </xsd:choice>
    </xsd:group>
    <xsd:element name="videoFile" type="CT_VideoFile"/>
    <xsd:simpleType name="ST_StyleMatrixColumnIndex">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FontCollectionIndex">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="major"/>
            <xsd:enumeration value="minor"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ColorSchemeIndex">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="dk1"/>
            <xsd:enumeration value="lt1"/>
            <xsd:enumeration value="dk2"/>
            <xsd:enumeration value="lt2"/>
            <xsd:enumeration value="accent1"/>
            <xsd:enumeration value="accent2"/>
            <xsd:enumeration value="accent3"/>
            <xsd:enumeration value="accent4"/>
```



```
            <xsd:enumeration value="accent5"/>
            <xsd:enumeration value="accent6"/>
            <xsd:enumeration value="hlink"/>
            <xsd:enumeration value="folHlink"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_ColorScheme">
        <xsd:sequence>
            <xsd:element name="dk1" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lt1" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="dk2" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lt2" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="accent1" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="accent2" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="accent3" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="accent4" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="accent5" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="accent6" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="hlink" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="folHlink" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomColor">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="optional" default=""/>
    </xsd:complexType>
    <xsd:complexType name="CT_SupplementalFont">
        <xsd:attribute name="script" type="xsd:string" use="required"/>
        <xsd:attribute name="typeface" type="ST_TextTypeface" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomColorList">
        <xsd:sequence>
            <xsd:element name="custClr" type="CT_CustomColor" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_FontCollection">
        <xsd:sequence>
            <xsd:element name="latin" type="CT_TextFont" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="ea" type="CT_TextFont" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cs" type="CT_TextFont" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="font" type="CT_SupplementalFont" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_EffectStyleItem">
        <xsd:sequence>
            <xsd:group ref="EG_EffectProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="scene3d" type="CT_Scene3D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sp3d" type="CT_Shape3D" minOccurs="0" maxOccurs="1"/>
```

```
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_FontScheme">
        <xsd:sequence>
            <xsd:element name="majorFont" type="CT_FontCollection" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="minorFont" type="CT_FontCollection" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FillStyleList">
        <xsd:sequence>
            <xsd:group ref="EG_FillProperties" minOccurs="3" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_LineStyleList">
        <xsd:sequence>
            <xsd:element name="ln" type="CT_LineProperties" minOccurs="3" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_EffectStyleList">
        <xsd:sequence>
            <xsd:element name="effectStyle" type="CT_EffectStyleItem" minOccurs="3"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_BackgroundFillStyleList">
        <xsd:sequence>
            <xsd:group ref="EG_FillProperties" minOccurs="3" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_StyleMatrix">
        <xsd:sequence>
            <xsd:element name="fillStyleLst" type="CT_FillStyleList" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lnStyleLst" type="CT_LineStyleList" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="effectStyleLst" type="CT_EffectStyleList" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="bgFillStyleLst" type="CT_BackgroundFillStyleList" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="optional" default=""/>
    </xsd:complexType>
    <xsd:complexType name="CT_BaseStyles">
        <xsd:sequence>
            <xsd:element name="clrScheme" type="CT_ColorScheme" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="fontScheme" type="CT_FontScheme" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="fmtScheme" type="CT_StyleMatrix" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_OfficeArtExtension">
        <xsd:sequence>
            <xsd:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
```



```
        <xsd:attribute name="uri" type="xsd:token" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Coordinate">
        <xsd:union memberTypes="ST_CoordinateUnqualified s:ST_UniversalMeasure"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_CoordinateUnqualified">
        <xsd:restriction base="xsd:long">
            <xsd:minInclusive value="-27273042329600"/>
            <xsd:maxInclusive value="27273042316900"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Coordinate32">
        <xsd:union memberTypes="ST_Coordinate32Unqualified s:ST_UniversalMeasure"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Coordinate32Unqualified">
        <xsd:restriction base="xsd:int"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PositiveCoordinate">
        <xsd:restriction base="xsd:long">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="27273042316900"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PositiveCoordinate32">
        <xsd:restriction base="ST_Coordinate32Unqualified">
            <xsd:minInclusive value="0"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Angle">
        <xsd:restriction base="xsd:int"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_Angle">
        <xsd:attribute name="val" type="ST_Angle" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_FixedAngle">
        <xsd:restriction base="ST_Angle">
            <xsd:minExclusive value="-5400000"/>
            <xsd:maxExclusive value="5400000"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PositiveFixedAngle">
        <xsd:restriction base="ST_Angle">
            <xsd:minInclusive value="0"/>
            <xsd:maxExclusive value="21600000"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PositiveFixedAngle">
        <xsd:attribute name="val" type="ST_PositiveFixedAngle" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Percentage">
        <xsd:union memberTypes="ST_PercentageDecimal s:ST_Percentage"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PercentageDecimal">
```

```
        <xsd:restriction base="xsd:int"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_Percentage">
        <xsd:attribute name="val" type="ST_Percentage" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PositivePercentage">
        <xsd:union memberTypes="ST_PositivePercentageDecimal s:ST_PositivePercentage"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PositivePercentageDecimal">
        <xsd:restriction base="ST_PercentageDecimal">
            <xsd:minInclusive value="0"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PositivePercentage">
        <xsd:attribute name="val" type="ST_PositivePercentage" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_FixedPercentage">
        <xsd:union memberTypes="ST_FixedPercentageDecimal s:ST_FixedPercentage"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FixedPercentageDecimal">
        <xsd:restriction base="ST_PercentageDecimal">
            <xsd:minInclusive value="-100000"/>
            <xsd:maxInclusive value="100000"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_FixedPercentage">
        <xsd:attribute name="val" type="ST_FixedPercentage" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PositiveFixedPercentage">
        <xsd:union memberTypes="ST_PositiveFixedPercentageDecimal s:ST_PositiveFixedPercentage"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PositiveFixedPercentageDecimal">
        <xsd:restriction base="ST_PercentageDecimal">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="100000"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PositiveFixedPercentage">
        <xsd:attribute name="val" type="ST_PositiveFixedPercentage" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Ratio">
        <xsd:attribute name="n" type="xsd:long" use="required"/>
        <xsd:attribute name="d" type="xsd:long" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Point2D">
        <xsd:attribute name="x" type="ST_Coordinate" use="required"/>
        <xsd:attribute name="y" type="ST_Coordinate" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PositiveSize2D">
        <xsd:attribute name="cx" type="ST_PositiveCoordinate" use="required"/>
        <xsd:attribute name="cy" type="ST_PositiveCoordinate" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ComplementTransform"/>
```



```
    <xsd:complexType name="CT_InverseTransform"/>
    <xsd:complexType name="CT_GrayscaleTransform"/>
    <xsd:complexType name="CT_GammaTransform"/>
    <xsd:complexType name="CT_InverseGammaTransform"/>
    <xsd:group name="EG_ColorTransform">
        <xsd:choice>
            <xsd:element name="tint" type="CT_PositiveFixedPercentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="shade" type="CT_PositiveFixedPercentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="comp" type="CT_ComplementTransform" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="inv" type="CT_InverseTransform" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="gray" type="CT_GrayscaleTransform" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="alpha" type="CT_PositiveFixedPercentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="alphaOff" type="CT_FixedPercentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="alphaMod" type="CT_PositivePercentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="hue" type="CT_PositiveFixedAngle" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="hueOff" type="CT_Angle" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="hueMod" type="CT_PositivePercentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="sat" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="satOff" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="satMod" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lum" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lumOff" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lumMod" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="red" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="redOff" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="redMod" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="green" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="greenOff" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="greenMod" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="blue" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="blueOff" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="blueMod" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="gamma" type="CT_GammaTransform" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="invGamma" type="CT_InverseGammaTransform" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_ScRgbColor">
        <xsd:sequence>
            <xsd:group ref="EG_ColorTransform" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="r" type="ST_Percentage" use="required"/>
        <xsd:attribute name="g" type="ST_Percentage" use="required"/>
        <xsd:attribute name="b" type="ST_Percentage" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SRgbColor">
        <xsd:sequence>
            <xsd:group ref="EG_ColorTransform" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="val" type="s:ST_HexColorRGB" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_HslColor">
        <xsd:sequence>
            <xsd:group ref="EG_ColorTransform" minOccurs="0" maxOccurs="unbounded"/>
```

```
        </xsd:sequence>
        <xsd:attribute name="hue" type="ST_PositiveFixedAngle" use="required"/>
        <xsd:attribute name="sat" type="ST_Percentage" use="required"/>
        <xsd:attribute name="lum" type="ST_Percentage" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SystemColorVal">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="scrollBar"/>
            <xsd:enumeration value="background"/>
            <xsd:enumeration value="activeCaption"/>
            <xsd:enumeration value="inactiveCaption"/>
            <xsd:enumeration value="menu"/>
            <xsd:enumeration value="window"/>
            <xsd:enumeration value="windowFrame"/>
            <xsd:enumeration value="menuText"/>
            <xsd:enumeration value="windowText"/>
            <xsd:enumeration value="captionText"/>
            <xsd:enumeration value="activeBorder"/>
            <xsd:enumeration value="inactiveBorder"/>
            <xsd:enumeration value="appWorkspace"/>
            <xsd:enumeration value="highlight"/>
            <xsd:enumeration value="highlightText"/>
            <xsd:enumeration value="btnFace"/>
            <xsd:enumeration value="btnShadow"/>
            <xsd:enumeration value="grayText"/>
            <xsd:enumeration value="btnText"/>
            <xsd:enumeration value="inactiveCaptionText"/>
            <xsd:enumeration value="btnHighlight"/>
            <xsd:enumeration value="3dDkShadow"/>
            <xsd:enumeration value="3dLight"/>
            <xsd:enumeration value="infoText"/>
            <xsd:enumeration value="infoBk"/>
            <xsd:enumeration value="hotLight"/>
            <xsd:enumeration value="gradientActiveCaption"/>
            <xsd:enumeration value="gradientInactiveCaption"/>
            <xsd:enumeration value="menuHighlight"/>
            <xsd:enumeration value="menuBar"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SystemColor">
        <xsd:sequence>
            <xsd:group ref="EG_ColorTransform" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="val" type="ST_SystemColorVal" use="required"/>
        <xsd:attribute name="lastClr" type="s:ST_HexColorRGB" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SchemeColorVal">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="bg1"/>
            <xsd:enumeration value="tx1"/>
            <xsd:enumeration value="bg2"/>
            <xsd:enumeration value="tx2"/>
            <xsd:enumeration value="accent1"/>
```



```
            <xsd:enumeration value="accent2"/>
            <xsd:enumeration value="accent3"/>
            <xsd:enumeration value="accent4"/>
            <xsd:enumeration value="accent5"/>
            <xsd:enumeration value="accent6"/>
            <xsd:enumeration value="hlink"/>
            <xsd:enumeration value="folHlink"/>
            <xsd:enumeration value="phClr"/>
            <xsd:enumeration value="dk1"/>
            <xsd:enumeration value="lt1"/>
            <xsd:enumeration value="dk2"/>
            <xsd:enumeration value="lt2"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SchemeColor">
        <xsd:sequence>
            <xsd:group ref="EG_ColorTransform" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="val" type="ST_SchemeColorVal" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PresetColorVal">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="aliceBlue"/>
            <xsd:enumeration value="antiqueWhite"/>
            <xsd:enumeration value="aqua"/>
            <xsd:enumeration value="aquamarine"/>
            <xsd:enumeration value="azure"/>
            <xsd:enumeration value="beige"/>
            <xsd:enumeration value="bisque"/>
            <xsd:enumeration value="black"/>
            <xsd:enumeration value="blanchedAlmond"/>
            <xsd:enumeration value="blue"/>
            <xsd:enumeration value="blueViolet"/>
            <xsd:enumeration value="brown"/>
            <xsd:enumeration value="burlyWood"/>
            <xsd:enumeration value="cadetBlue"/>
            <xsd:enumeration value="chartreuse"/>
            <xsd:enumeration value="chocolate"/>
            <xsd:enumeration value="coral"/>
            <xsd:enumeration value="cornflowerBlue"/>
            <xsd:enumeration value="cornsilk"/>
            <xsd:enumeration value="crimson"/>
            <xsd:enumeration value="cyan"/>
            <xsd:enumeration value="darkBlue"/>
            <xsd:enumeration value="darkCyan"/>
            <xsd:enumeration value="darkGoldenrod"/>
            <xsd:enumeration value="darkGray"/>
            <xsd:enumeration value="darkGrey"/>
            <xsd:enumeration value="darkGreen"/>
            <xsd:enumeration value="darkKhaki"/>
            <xsd:enumeration value="darkMagenta"/>
            <xsd:enumeration value="darkOliveGreen"/>
            <xsd:enumeration value="darkOrange"/>
```

```
            <xsd:enumeration value="darkOrchid"/>
            <xsd:enumeration value="darkRed"/>
            <xsd:enumeration value="darkSalmon"/>
            <xsd:enumeration value="darkSeaGreen"/>
            <xsd:enumeration value="darkSlateBlue"/>
            <xsd:enumeration value="darkSlateGray"/>
            <xsd:enumeration value="darkSlateGrey"/>
            <xsd:enumeration value="darkTurquoise"/>
            <xsd:enumeration value="darkViolet"/>
            <xsd:enumeration value="dkBlue"/>
            <xsd:enumeration value="dkCyan"/>
            <xsd:enumeration value="dkGoldenrod"/>
            <xsd:enumeration value="dkGray"/>
            <xsd:enumeration value="dkGrey"/>
            <xsd:enumeration value="dkGreen"/>
            <xsd:enumeration value="dkKhaki"/>
            <xsd:enumeration value="dkMagenta"/>
            <xsd:enumeration value="dkOliveGreen"/>
            <xsd:enumeration value="dkOrange"/>
            <xsd:enumeration value="dkOrchid"/>
            <xsd:enumeration value="dkRed"/>
            <xsd:enumeration value="dkSalmon"/>
            <xsd:enumeration value="dkSeaGreen"/>
            <xsd:enumeration value="dkSlateBlue"/>
            <xsd:enumeration value="dkSlateGray"/>
            <xsd:enumeration value="dkSlateGrey"/>
            <xsd:enumeration value="dkTurquoise"/>
            <xsd:enumeration value="dkViolet"/>
            <xsd:enumeration value="deepPink"/>
            <xsd:enumeration value="deepSkyBlue"/>
            <xsd:enumeration value="dimGray"/>
            <xsd:enumeration value="dimGrey"/>
            <xsd:enumeration value="dodgerBlue"/>
            <xsd:enumeration value="firebrick"/>
            <xsd:enumeration value="floralWhite"/>
            <xsd:enumeration value="forestGreen"/>
            <xsd:enumeration value="fuchsia"/>
            <xsd:enumeration value="gainsboro"/>
            <xsd:enumeration value="ghostWhite"/>
            <xsd:enumeration value="gold"/>
            <xsd:enumeration value="goldenrod"/>
            <xsd:enumeration value="gray"/>
            <xsd:enumeration value="grey"/>
            <xsd:enumeration value="green"/>
            <xsd:enumeration value="greenYellow"/>
            <xsd:enumeration value="honeydew"/>
            <xsd:enumeration value="hotPink"/>
            <xsd:enumeration value="indianRed"/>
            <xsd:enumeration value="indigo"/>
            <xsd:enumeration value="ivory"/>
            <xsd:enumeration value="khaki"/>
            <xsd:enumeration value="lavender"/>
            <xsd:enumeration value="lavenderBlush"/>
```



```
            <xsd:enumeration value="lawnGreen"/>
            <xsd:enumeration value="lemonChiffon"/>
            <xsd:enumeration value="lightBlue"/>
            <xsd:enumeration value="lightCoral"/>
            <xsd:enumeration value="lightCyan"/>
            <xsd:enumeration value="lightGoldenrodYellow"/>
            <xsd:enumeration value="lightGray"/>
            <xsd:enumeration value="lightGrey"/>
            <xsd:enumeration value="lightGreen"/>
            <xsd:enumeration value="lightPink"/>
            <xsd:enumeration value="lightSalmon"/>
            <xsd:enumeration value="lightSeaGreen"/>
            <xsd:enumeration value="lightSkyBlue"/>
            <xsd:enumeration value="lightSlateGray"/>
            <xsd:enumeration value="lightSlateGrey"/>
            <xsd:enumeration value="lightSteelBlue"/>
            <xsd:enumeration value="lightYellow"/>
            <xsd:enumeration value="ltBlue"/>
            <xsd:enumeration value="ltCoral"/>
            <xsd:enumeration value="ltCyan"/>
            <xsd:enumeration value="ltGoldenrodYellow"/>
            <xsd:enumeration value="ltGray"/>
            <xsd:enumeration value="ltGrey"/>
            <xsd:enumeration value="ltGreen"/>
            <xsd:enumeration value="ltPink"/>
            <xsd:enumeration value="ltSalmon"/>
            <xsd:enumeration value="ltSeaGreen"/>
            <xsd:enumeration value="ltSkyBlue"/>
            <xsd:enumeration value="ltSlateGray"/>
            <xsd:enumeration value="ltSlateGrey"/>
            <xsd:enumeration value="ltSteelBlue"/>
            <xsd:enumeration value="ltYellow"/>
            <xsd:enumeration value="lime"/>
            <xsd:enumeration value="limeGreen"/>
            <xsd:enumeration value="linen"/>
            <xsd:enumeration value="magenta"/>
            <xsd:enumeration value="maroon"/>
            <xsd:enumeration value="medAquamarine"/>
            <xsd:enumeration value="medBlue"/>
            <xsd:enumeration value="medOrchid"/>
            <xsd:enumeration value="medPurple"/>
            <xsd:enumeration value="medSeaGreen"/>
            <xsd:enumeration value="medSlateBlue"/>
            <xsd:enumeration value="medSpringGreen"/>
            <xsd:enumeration value="medTurquoise"/>
            <xsd:enumeration value="medVioletRed"/>
            <xsd:enumeration value="mediumAquamarine"/>
            <xsd:enumeration value="mediumBlue"/>
            <xsd:enumeration value="mediumOrchid"/>
            <xsd:enumeration value="mediumPurple"/>
            <xsd:enumeration value="mediumSeaGreen"/>
            <xsd:enumeration value="mediumSlateBlue"/>
            <xsd:enumeration value="mediumSpringGreen"/>
```

```
            <xsd:enumeration value="mediumTurquoise"/>
            <xsd:enumeration value="mediumVioletRed"/>
            <xsd:enumeration value="midnightBlue"/>
            <xsd:enumeration value="mintCream"/>
            <xsd:enumeration value="mistyRose"/>
            <xsd:enumeration value="moccasin"/>
            <xsd:enumeration value="navajoWhite"/>
            <xsd:enumeration value="navy"/>
            <xsd:enumeration value="oldLace"/>
            <xsd:enumeration value="olive"/>
            <xsd:enumeration value="oliveDrab"/>
            <xsd:enumeration value="orange"/>
            <xsd:enumeration value="orangeRed"/>
            <xsd:enumeration value="orchid"/>
            <xsd:enumeration value="paleGoldenrod"/>
            <xsd:enumeration value="paleGreen"/>
            <xsd:enumeration value="paleTurquoise"/>
            <xsd:enumeration value="paleVioletRed"/>
            <xsd:enumeration value="papayaWhip"/>
            <xsd:enumeration value="peachPuff"/>
            <xsd:enumeration value="peru"/>
            <xsd:enumeration value="pink"/>
            <xsd:enumeration value="plum"/>
            <xsd:enumeration value="powderBlue"/>
            <xsd:enumeration value="purple"/>
            <xsd:enumeration value="red"/>
            <xsd:enumeration value="rosyBrown"/>
            <xsd:enumeration value="royalBlue"/>
            <xsd:enumeration value="saddleBrown"/>
            <xsd:enumeration value="salmon"/>
            <xsd:enumeration value="sandyBrown"/>
            <xsd:enumeration value="seaGreen"/>
            <xsd:enumeration value="seaShell"/>
            <xsd:enumeration value="sienna"/>
            <xsd:enumeration value="silver"/>
            <xsd:enumeration value="skyBlue"/>
            <xsd:enumeration value="slateBlue"/>
            <xsd:enumeration value="slateGray"/>
            <xsd:enumeration value="slateGrey"/>
            <xsd:enumeration value="snow"/>
            <xsd:enumeration value="springGreen"/>
            <xsd:enumeration value="steelBlue"/>
            <xsd:enumeration value="tan"/>
            <xsd:enumeration value="teal"/>
            <xsd:enumeration value="thistle"/>
            <xsd:enumeration value="tomato"/>
            <xsd:enumeration value="turquoise"/>
            <xsd:enumeration value="violet"/>
            <xsd:enumeration value="wheat"/>
            <xsd:enumeration value="white"/>
            <xsd:enumeration value="whiteSmoke"/>
            <xsd:enumeration value="yellow"/>
            <xsd:enumeration value="yellowGreen"/>
```



```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PresetColor">
        <xsd:sequence>
            <xsd:group ref="EG_ColorTransform" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="val" type="ST_PresetColorVal" use="required"/>
    </xsd:complexType>
    <xsd:group name="EG_OfficeArtExtensionList">
        <xsd:sequence>
            <xsd:element name="ext" type="CT_OfficeArtExtension" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_OfficeArtExtensionList">
        <xsd:sequence>
            <xsd:group ref="EG_OfficeArtExtensionList" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Scale2D">
        <xsd:sequence>
            <xsd:element name="sx" type="CT_Ratio" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="sy" type="CT_Ratio" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Transform2D">
        <xsd:sequence>
            <xsd:element name="off" type="CT_Point2D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ext" type="CT_PositiveSize2D" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="rot" type="ST_Angle" use="optional" default="0"/>
        <xsd:attribute name="flipH" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="flipV" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupTransform2D">
        <xsd:sequence>
            <xsd:element name="off" type="CT_Point2D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ext" type="CT_PositiveSize2D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="chOff" type="CT_Point2D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="chExt" type="CT_PositiveSize2D" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="rot" type="ST_Angle" use="optional" default="0"/>
        <xsd:attribute name="flipH" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="flipV" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Point3D">
        <xsd:attribute name="x" type="ST_Coordinate" use="required"/>
        <xsd:attribute name="y" type="ST_Coordinate" use="required"/>
        <xsd:attribute name="z" type="ST_Coordinate" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Vector3D">
        <xsd:attribute name="dx" type="ST_Coordinate" use="required"/>
        <xsd:attribute name="dy" type="ST_Coordinate" use="required"/>
        <xsd:attribute name="dz" type="ST_Coordinate" use="required"/>
```

```
    </xsd:complexType>
    <xsd:complexType name="CT_SphereCoords">
        <xsd:attribute name="lat" type="ST_PositiveFixedAngle" use="required"/>
        <xsd:attribute name="lon" type="ST_PositiveFixedAngle" use="required"/>
        <xsd:attribute name="rev" type="ST_PositiveFixedAngle" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RelativeRect">
        <xsd:attribute name="l" type="ST_Percentage" use="optional" default="0%"/>
        <xsd:attribute name="t" type="ST_Percentage" use="optional" default="0%"/>
        <xsd:attribute name="r" type="ST_Percentage" use="optional" default="0%"/>
        <xsd:attribute name="b" type="ST_Percentage" use="optional" default="0%"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_RectAlignment">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="tl"/>
            <xsd:enumeration value="t"/>
            <xsd:enumeration value="tr"/>
            <xsd:enumeration value="l"/>
            <xsd:enumeration value="ctr"/>
            <xsd:enumeration value="r"/>
            <xsd:enumeration value="bl"/>
            <xsd:enumeration value="b"/>
            <xsd:enumeration value="br"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:group name="EG_ColorChoice">
        <xsd:choice>
            <xsd:element name="scrgbClr" type="CT_ScRgbColor" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="srgbClr" type="CT_SRgbColor" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="hslClr" type="CT_HslColor" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="sysClr" type="CT_SystemColor" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="schemeClr" type="CT_SchemeColor" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="prstClr" type="CT_PresetColor" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_Color">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ColorMRU">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_BlackWhiteMode">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="clr"/>
            <xsd:enumeration value="auto"/>
            <xsd:enumeration value="gray"/>
            <xsd:enumeration value="ltGray"/>
            <xsd:enumeration value="invGray"/>
            <xsd:enumeration value="grayWhite"/>
```



```
            <xsd:enumeration value="blackGray"/>
            <xsd:enumeration value="blackWhite"/>
            <xsd:enumeration value="black"/>
            <xsd:enumeration value="white"/>
            <xsd:enumeration value="hidden"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:attributeGroup name="AG_Blob">
        <xsd:attribute ref="r:embed" use="optional" default=""/>
        <xsd:attribute ref="r:link" use="optional" default=""/>
    </xsd:attributeGroup>
    <xsd:complexType name="CT_EmbeddedWAVAudioFile">
        <xsd:attribute ref="r:embed" use="required"/>
        <xsd:attribute name="name" type="xsd:string" use="optional" default=""/>
    </xsd:complexType>
    <xsd:complexType name="CT_Hyperlink">
        <xsd:sequence>
            <xsd:element name="snd" type="CT_EmbeddedWAVAudioFile" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute ref="r:id" use="optional"/>
        <xsd:attribute name="invalidUrl" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="action" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="tgtFrame" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="tooltip" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="history" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="highlightClick" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="endSnd" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DrawingElementId">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:attributeGroup name="AG_Locking">
        <xsd:attribute name="noGrp" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noSelect" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noRot" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noChangeAspect" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noMove" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noResize" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noEditPoints" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noAdjustHandles" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noChangeArrowheads" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noChangeShapeType" type="xsd:boolean" use="optional" default="false"/>
    </xsd:attributeGroup>
    <xsd:complexType name="CT_ConnectorLocking">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_Locking"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ShapeLocking">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
```

```
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_Locking"/>
        <xsd:attribute name="noTextEdit" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PictureLocking">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_Locking"/>
        <xsd:attribute name="noCrop" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupLocking">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="noGrp" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noUngrp" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noSelect" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noRot" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noChangeAspect" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noMove" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noResize" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GraphicalObjectFrameLocking">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="noGrp" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noDrilldown" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noSelect" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noChangeAspect" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noMove" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noResize" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ContentPartLocking">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_Locking"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NonVisualDrawingProps">
        <xsd:sequence>
            <xsd:element name="hlinkClick" type="CT_Hyperlink" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="hlinkHover" type="CT_Hyperlink" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="ST_DrawingElementId" use="required"/>
        <xsd:attribute name="name" type="xsd:string" use="required"/>
        <xsd:attribute name="descr" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="hidden" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="title" type="xsd:string" use="optional" default=""/>
    </xsd:complexType>
    <xsd:complexType name="CT_NonVisualDrawingShapeProps">
```



```
        <xsd:sequence>
            <xsd:element name="spLocks" type="CT_ShapeLocking" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="txBox" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NonVisualConnectorProperties">
        <xsd:sequence>
            <xsd:element name="cxnSpLocks" type="CT_ConnectorLocking" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="stCxn" type="CT_Connection" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="endCxn" type="CT_Connection" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_NonVisualPictureProperties">
        <xsd:sequence>
            <xsd:element name="picLocks" type="CT_PictureLocking" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="preferRelativeResize" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NonVisualGroupDrawingShapeProps">
        <xsd:sequence>
            <xsd:element name="grpSpLocks" type="CT_GroupLocking" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_NonVisualGraphicFrameProperties">
        <xsd:sequence>
            <xsd:element name="graphicFrameLocks" type="CT_GraphicalObjectFrameLocking" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_NonVisualContentPartProperties">
        <xsd:sequence>
            <xsd:element name="cpLocks" type="CT_ContentPartLocking" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="isComment" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GraphicalObjectData">
        <xsd:sequence>
            <xsd:any minOccurs="0" maxOccurs="unbounded" processContents="strict"/>
        </xsd:sequence>
        <xsd:attribute name="uri" type="xsd:token" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GraphicalObject">
        <xsd:sequence>
            <xsd:element name="graphicData" type="CT_GraphicalObjectData"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="graphic" type="CT_GraphicalObject"/>
```

```
    <xsd:simpleType name="ST_ChartBuildStep">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="category"/>
            <xsd:enumeration value="ptInCategory"/>
            <xsd:enumeration value="series"/>
            <xsd:enumeration value="ptInSeries"/>
            <xsd:enumeration value="allPts"/>
            <xsd:enumeration value="gridLegend"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_DgmBuildStep">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="sp"/>
            <xsd:enumeration value="bg"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_AnimationDgmElement">
        <xsd:attribute name="id" type="s:ST_Guid" use="optional" default="{00000000-0000-0000-0000-
          000000000000}"/>
        <xsd:attribute name="bldStep" type="ST_DgmBuildStep" use="optional" default="sp"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AnimationChartElement">
        <xsd:attribute name="seriesIdx" type="xsd:int" use="optional" default="-1"/>
        <xsd:attribute name="categoryIdx" type="xsd:int" use="optional" default="-1"/>
        <xsd:attribute name="bldStep" type="ST_ChartBuildStep" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AnimationElementChoice">
        <xsd:choice minOccurs="1" maxOccurs="1">
            <xsd:element name="dgm" type="CT_AnimationDgmElement"/>
            <xsd:element name="chart" type="CT_AnimationChartElement"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:simpleType name="ST_AnimationBuildType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="allAtOnce"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_AnimationDgmOnlyBuildType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="one"/>
            <xsd:enumeration value="lvlOne"/>
            <xsd:enumeration value="lvlAtOnce"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_AnimationDgmBuildType">
        <xsd:union memberTypes="ST_AnimationBuildType ST_AnimationDgmOnlyBuildType"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_AnimationDgmBuildProperties">
        <xsd:attribute name="bld" type="ST_AnimationDgmBuildType" use="optional" default="allAtOnce"/>
        <xsd:attribute name="rev" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_AnimationChartOnlyBuildType">
        <xsd:restriction base="xsd:token">
```



```
            <xsd:enumeration value="series"/>
            <xsd:enumeration value="category"/>
            <xsd:enumeration value="seriesEl"/>
            <xsd:enumeration value="categoryEl"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_AnimationChartBuildType">
        <xsd:union memberTypes="ST_AnimationBuildType ST_AnimationChartOnlyBuildType"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_AnimationChartBuildProperties">
        <xsd:attribute name="bld" type="ST_AnimationChartBuildType" use="optional"
          default="allAtOnce"/>
        <xsd:attribute name="animBg" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AnimationGraphicalObjectBuildProperties">
        <xsd:choice>
            <xsd:element name="bldDgm" type="CT_AnimationDgmBuildProperties"/>
            <xsd:element name="bldChart" type="CT_AnimationChartBuildProperties"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_BackgroundFormatting">
        <xsd:sequence>
            <xsd:group ref="EG_FillProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_EffectProperties" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_WholeE2oFormatting">
        <xsd:sequence>
            <xsd:element name="ln" type="CT_LineProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_EffectProperties" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlUseShapeRectangle"/>
    <xsd:complexType name="CT_GvmlTextShape">
        <xsd:sequence>
            <xsd:element name="txBody" type="CT_TextBody" minOccurs="1" maxOccurs="1"/>
            <xsd:choice>
                <xsd:element name="useSpRect" type="CT_GvmlUseShapeRectangle" minOccurs="1"
                  maxOccurs="1"/>
                <xsd:element name="xfrm" type="CT_Transform2D" minOccurs="1" maxOccurs="1"/>
            </xsd:choice>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlShapeNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvSpPr" type="CT_NonVisualDrawingShapeProps" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlShape">
        <xsd:sequence>
```

```
            <xsd:element name="nvSpPr" type="CT_GvmlShapeNonVisual" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="spPr" type="CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="txSp" type="CT_GvmlTextShape" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="style" type="CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlConnectorNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvCxnSpPr" type="CT_NonVisualConnectorProperties" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlConnector">
        <xsd:sequence>
            <xsd:element name="nvCxnSpPr" type="CT_GvmlConnectorNonVisual" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="spPr" type="CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="style" type="CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlPictureNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvPicPr" type="CT_NonVisualPictureProperties" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlPicture">
        <xsd:sequence>
            <xsd:element name="nvPicPr" type="CT_GvmlPictureNonVisual" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="blipFill" type="CT_BlipFillProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="spPr" type="CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="style" type="CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlGraphicFrameNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvGraphicFramePr" type="CT_NonVisualGraphicFrameProperties"
              minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlGraphicalObjectFrame">
        <xsd:sequence>
            <xsd:element name="nvGraphicFramePr" type="CT_GvmlGraphicFrameNonVisual" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element ref="graphic" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="xfrm" type="CT_Transform2D" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
```



```
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlGroupShapeNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvGrpSpPr" type="CT_NonVisualGroupDrawingShapeProps" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlGroupShape">
        <xsd:sequence>
            <xsd:element name="nvGrpSpPr" type="CT_GvmlGroupShapeNonVisual" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="grpSpPr" type="CT_GroupShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="txSp" type="CT_GvmlTextShape"/>
                <xsd:element name="sp" type="CT_GvmlShape"/>
                <xsd:element name="cxnSp" type="CT_GvmlConnector"/>
                <xsd:element name="pic" type="CT_GvmlPicture"/>
                <xsd:element name="graphicFrame" type="CT_GvmlGraphicalObjectFrame"/>
                <xsd:element name="grpSp" type="CT_GvmlGroupShape"/>
            </xsd:choice>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_PresetCameraType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="legacyObliqueTopLeft"/>
            <xsd:enumeration value="legacyObliqueTop"/>
            <xsd:enumeration value="legacyObliqueTopRight"/>
            <xsd:enumeration value="legacyObliqueLeft"/>
            <xsd:enumeration value="legacyObliqueFront"/>
            <xsd:enumeration value="legacyObliqueRight"/>
            <xsd:enumeration value="legacyObliqueBottomLeft"/>
            <xsd:enumeration value="legacyObliqueBottom"/>
            <xsd:enumeration value="legacyObliqueBottomRight"/>
            <xsd:enumeration value="legacyPerspectiveTopLeft"/>
            <xsd:enumeration value="legacyPerspectiveTop"/>
            <xsd:enumeration value="legacyPerspectiveTopRight"/>
            <xsd:enumeration value="legacyPerspectiveLeft"/>
            <xsd:enumeration value="legacyPerspectiveFront"/>
            <xsd:enumeration value="legacyPerspectiveRight"/>
            <xsd:enumeration value="legacyPerspectiveBottomLeft"/>
            <xsd:enumeration value="legacyPerspectiveBottom"/>
            <xsd:enumeration value="legacyPerspectiveBottomRight"/>
            <xsd:enumeration value="orthographicFront"/>
            <xsd:enumeration value="isometricTopUp"/>
            <xsd:enumeration value="isometricTopDown"/>
            <xsd:enumeration value="isometricBottomUp"/>
            <xsd:enumeration value="isometricBottomDown"/>
            <xsd:enumeration value="isometricLeftUp"/>
            <xsd:enumeration value="isometricLeftDown"/>
            <xsd:enumeration value="isometricRightUp"/>
```

```
            <xsd:enumeration value="isometricRightDown"/>
            <xsd:enumeration value="isometricOffAxis1Left"/>
            <xsd:enumeration value="isometricOffAxis1Right"/>
            <xsd:enumeration value="isometricOffAxis1Top"/>
            <xsd:enumeration value="isometricOffAxis2Left"/>
            <xsd:enumeration value="isometricOffAxis2Right"/>
            <xsd:enumeration value="isometricOffAxis2Top"/>
            <xsd:enumeration value="isometricOffAxis3Left"/>
            <xsd:enumeration value="isometricOffAxis3Right"/>
            <xsd:enumeration value="isometricOffAxis3Bottom"/>
            <xsd:enumeration value="isometricOffAxis4Left"/>
            <xsd:enumeration value="isometricOffAxis4Right"/>
            <xsd:enumeration value="isometricOffAxis4Bottom"/>
            <xsd:enumeration value="obliqueTopLeft"/>
            <xsd:enumeration value="obliqueTop"/>
            <xsd:enumeration value="obliqueTopRight"/>
            <xsd:enumeration value="obliqueLeft"/>
            <xsd:enumeration value="obliqueRight"/>
            <xsd:enumeration value="obliqueBottomLeft"/>
            <xsd:enumeration value="obliqueBottom"/>
            <xsd:enumeration value="obliqueBottomRight"/>
            <xsd:enumeration value="perspectiveFront"/>
            <xsd:enumeration value="perspectiveLeft"/>
            <xsd:enumeration value="perspectiveRight"/>
            <xsd:enumeration value="perspectiveAbove"/>
            <xsd:enumeration value="perspectiveBelow"/>
            <xsd:enumeration value="perspectiveAboveLeftFacing"/>
            <xsd:enumeration value="perspectiveAboveRightFacing"/>
            <xsd:enumeration value="perspectiveContrastingLeftFacing"/>
            <xsd:enumeration value="perspectiveContrastingRightFacing"/>
            <xsd:enumeration value="perspectiveHeroicLeftFacing"/>
            <xsd:enumeration value="perspectiveHeroicRightFacing"/>
            <xsd:enumeration value="perspectiveHeroicExtremeLeftFacing"/>
            <xsd:enumeration value="perspectiveHeroicExtremeRightFacing"/>
            <xsd:enumeration value="perspectiveRelaxed"/>
            <xsd:enumeration value="perspectiveRelaxedModerately"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FOVAngle">
        <xsd:restriction base="ST_Angle">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="10800000"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Camera">
        <xsd:sequence>
            <xsd:element name="rot" type="CT_SphereCoords" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="prst" type="ST_PresetCameraType" use="required"/>
        <xsd:attribute name="fov" type="ST_FOVAngle" use="optional"/>
        <xsd:attribute name="zoom" type="ST_PositivePercentage" use="optional" default="100%"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_LightRigDirection">
```



```
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="tl"/>
            <xsd:enumeration value="t"/>
            <xsd:enumeration value="tr"/>
            <xsd:enumeration value="l"/>
            <xsd:enumeration value="r"/>
            <xsd:enumeration value="bl"/>
            <xsd:enumeration value="b"/>
            <xsd:enumeration value="br"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_LightRigType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="legacyFlat1"/>
            <xsd:enumeration value="legacyFlat2"/>
            <xsd:enumeration value="legacyFlat3"/>
            <xsd:enumeration value="legacyFlat4"/>
            <xsd:enumeration value="legacyNormal1"/>
            <xsd:enumeration value="legacyNormal2"/>
            <xsd:enumeration value="legacyNormal3"/>
            <xsd:enumeration value="legacyNormal4"/>
            <xsd:enumeration value="legacyHarsh1"/>
            <xsd:enumeration value="legacyHarsh2"/>
            <xsd:enumeration value="legacyHarsh3"/>
            <xsd:enumeration value="legacyHarsh4"/>
            <xsd:enumeration value="threePt"/>
            <xsd:enumeration value="balanced"/>
            <xsd:enumeration value="soft"/>
            <xsd:enumeration value="harsh"/>
            <xsd:enumeration value="flood"/>
            <xsd:enumeration value="contrasting"/>
            <xsd:enumeration value="morning"/>
            <xsd:enumeration value="sunrise"/>
            <xsd:enumeration value="sunset"/>
            <xsd:enumeration value="chilly"/>
            <xsd:enumeration value="freezing"/>
            <xsd:enumeration value="flat"/>
            <xsd:enumeration value="twoPt"/>
            <xsd:enumeration value="glow"/>
            <xsd:enumeration value="brightRoom"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_LightRig">
        <xsd:sequence>
            <xsd:element name="rot" type="CT_SphereCoords" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="rig" type="ST_LightRigType" use="required"/>
        <xsd:attribute name="dir" type="ST_LightRigDirection" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Scene3D">
        <xsd:sequence>
            <xsd:element name="camera" type="CT_Camera" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lightRig" type="CT_LightRig" minOccurs="1" maxOccurs="1"/>
```

```
            <xsd:element name="backdrop" type="CT_Backdrop" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Backdrop">
        <xsd:sequence>
            <xsd:element name="anchor" type="CT_Point3D" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="norm" type="CT_Vector3D" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="up" type="CT_Vector3D" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_BevelPresetType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="relaxedInset"/>
            <xsd:enumeration value="circle"/>
            <xsd:enumeration value="slope"/>
            <xsd:enumeration value="cross"/>
            <xsd:enumeration value="angle"/>
            <xsd:enumeration value="softRound"/>
            <xsd:enumeration value="convex"/>
            <xsd:enumeration value="coolSlant"/>
            <xsd:enumeration value="divot"/>
            <xsd:enumeration value="riblet"/>
            <xsd:enumeration value="hardEdge"/>
            <xsd:enumeration value="artDeco"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Bevel">
        <xsd:attribute name="w" type="ST_PositiveCoordinate" use="optional" default="76200"/>
        <xsd:attribute name="h" type="ST_PositiveCoordinate" use="optional" default="76200"/>
        <xsd:attribute name="prst" type="ST_BevelPresetType" use="optional" default="circle"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PresetMaterialType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="legacyMatte"/>
            <xsd:enumeration value="legacyPlastic"/>
            <xsd:enumeration value="legacyMetal"/>
            <xsd:enumeration value="legacyWireframe"/>
            <xsd:enumeration value="matte"/>
            <xsd:enumeration value="plastic"/>
            <xsd:enumeration value="metal"/>
            <xsd:enumeration value="warmMatte"/>
            <xsd:enumeration value="translucentPowder"/>
            <xsd:enumeration value="powder"/>
            <xsd:enumeration value="dkEdge"/>
            <xsd:enumeration value="softEdge"/>
            <xsd:enumeration value="clear"/>
            <xsd:enumeration value="flat"/>
            <xsd:enumeration value="softmetal"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Shape3D">
```



```
        <xsd:sequence>
            <xsd:element name="bevelT" type="CT_Bevel" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bevelB" type="CT_Bevel" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extrusionClr" type="CT_Color" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="contourClr" type="CT_Color" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="z" type="ST_Coordinate" use="optional" default="0"/>
        <xsd:attribute name="extrusionH" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="contourW" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="prstMaterial" type="ST_PresetMaterialType" use="optional"
          default="warmMatte"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FlatText">
        <xsd:attribute name="z" type="ST_Coordinate" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:group name="EG_Text3D">
        <xsd:choice>
            <xsd:element name="sp3d" type="CT_Shape3D" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="flatTx" type="CT_FlatText" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_AlphaBiLevelEffect">
        <xsd:attribute name="thresh" type="ST_PositiveFixedPercentage" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AlphaCeilingEffect"/>
    <xsd:complexType name="CT_AlphaFloorEffect"/>
    <xsd:complexType name="CT_AlphaInverseEffect">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_AlphaModulateFixedEffect">
        <xsd:attribute name="amt" type="ST_PositivePercentage" use="optional" default="100%"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AlphaOutsetEffect">
        <xsd:attribute name="rad" type="ST_Coordinate" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AlphaReplaceEffect">
        <xsd:attribute name="a" type="ST_PositiveFixedPercentage" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_BiLevelEffect">
        <xsd:attribute name="thresh" type="ST_PositiveFixedPercentage" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_BlurEffect">
        <xsd:attribute name="rad" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="grow" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ColorChangeEffect">
        <xsd:sequence>
            <xsd:element name="clrFrom" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="clrTo" type="CT_Color" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
```

```
        <xsd:attribute name="useA" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ColorReplaceEffect">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DuotoneEffect">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="2" maxOccurs="2"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GlowEffect">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="rad" type="ST_PositiveCoordinate" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GrayscaleEffect"/>
    <xsd:complexType name="CT_HSLEffect">
        <xsd:attribute name="hue" type="ST_PositiveFixedAngle" use="optional" default="0"/>
        <xsd:attribute name="sat" type="ST_FixedPercentage" use="optional" default="0%"/>
        <xsd:attribute name="lum" type="ST_FixedPercentage" use="optional" default="0%"/>
    </xsd:complexType>
    <xsd:complexType name="CT_InnerShadowEffect">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="blurRad" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="dist" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="dir" type="ST_PositiveFixedAngle" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_LuminanceEffect">
        <xsd:attribute name="bright" type="ST_FixedPercentage" use="optional" default="0%"/>
        <xsd:attribute name="contrast" type="ST_FixedPercentage" use="optional" default="0%"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OuterShadowEffect">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="blurRad" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="dist" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="dir" type="ST_PositiveFixedAngle" use="optional" default="0"/>
        <xsd:attribute name="sx" type="ST_Percentage" use="optional" default="100%"/>
        <xsd:attribute name="sy" type="ST_Percentage" use="optional" default="100%"/>
        <xsd:attribute name="kx" type="ST_FixedAngle" use="optional" default="0"/>
        <xsd:attribute name="ky" type="ST_FixedAngle" use="optional" default="0"/>
        <xsd:attribute name="algn" type="ST_RectAlignment" use="optional" default="b"/>
        <xsd:attribute name="rotWithShape" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PresetShadowVal">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="shdw1"/>
```



```
            <xsd:enumeration value="shdw2"/>
            <xsd:enumeration value="shdw3"/>
            <xsd:enumeration value="shdw4"/>
            <xsd:enumeration value="shdw5"/>
            <xsd:enumeration value="shdw6"/>
            <xsd:enumeration value="shdw7"/>
            <xsd:enumeration value="shdw8"/>
            <xsd:enumeration value="shdw9"/>
            <xsd:enumeration value="shdw10"/>
            <xsd:enumeration value="shdw11"/>
            <xsd:enumeration value="shdw12"/>
            <xsd:enumeration value="shdw13"/>
            <xsd:enumeration value="shdw14"/>
            <xsd:enumeration value="shdw15"/>
            <xsd:enumeration value="shdw16"/>
            <xsd:enumeration value="shdw17"/>
            <xsd:enumeration value="shdw18"/>
            <xsd:enumeration value="shdw19"/>
            <xsd:enumeration value="shdw20"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PresetShadowEffect">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="prst" type="ST_PresetShadowVal" use="required"/>
        <xsd:attribute name="dist" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="dir" type="ST_PositiveFixedAngle" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ReflectionEffect">
        <xsd:attribute name="blurRad" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="stA" type="ST_PositiveFixedPercentage" use="optional" default="100%"/>
        <xsd:attribute name="stPos" type="ST_PositiveFixedPercentage" use="optional" default="0%"/>
        <xsd:attribute name="endA" type="ST_PositiveFixedPercentage" use="optional" default="0%"/>
        <xsd:attribute name="endPos" type="ST_PositiveFixedPercentage" use="optional" default="100%"/>
        <xsd:attribute name="dist" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="dir" type="ST_PositiveFixedAngle" use="optional" default="0"/>
        <xsd:attribute name="fadeDir" type="ST_PositiveFixedAngle" use="optional" default="5400000"/>
        <xsd:attribute name="sx" type="ST_Percentage" use="optional" default="100%"/>
        <xsd:attribute name="sy" type="ST_Percentage" use="optional" default="100%"/>
        <xsd:attribute name="kx" type="ST_FixedAngle" use="optional" default="0"/>
        <xsd:attribute name="ky" type="ST_FixedAngle" use="optional" default="0"/>
        <xsd:attribute name="algn" type="ST_RectAlignment" use="optional" default="b"/>
        <xsd:attribute name="rotWithShape" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RelativeOffsetEffect">
        <xsd:attribute name="tx" type="ST_Percentage" use="optional" default="0%"/>
        <xsd:attribute name="ty" type="ST_Percentage" use="optional" default="0%"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SoftEdgesEffect">
        <xsd:attribute name="rad" type="ST_PositiveCoordinate" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TintEffect">
```

```
        <xsd:attribute name="hue" type="ST_PositiveFixedAngle" use="optional" default="0"/>
        <xsd:attribute name="amt" type="ST_FixedPercentage" use="optional" default="0%"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TransformEffect">
        <xsd:attribute name="sx" type="ST_Percentage" use="optional" default="100%"/>
        <xsd:attribute name="sy" type="ST_Percentage" use="optional" default="100%"/>
        <xsd:attribute name="kx" type="ST_FixedAngle" use="optional" default="0"/>
        <xsd:attribute name="ky" type="ST_FixedAngle" use="optional" default="0"/>
        <xsd:attribute name="tx" type="ST_Coordinate" use="optional" default="0"/>
        <xsd:attribute name="ty" type="ST_Coordinate" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NoFillProperties"/>
    <xsd:complexType name="CT_SolidColorFillProperties">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_LinearShadeProperties">
        <xsd:attribute name="ang" type="ST_PositiveFixedAngle" use="optional"/>
        <xsd:attribute name="scaled" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PathShadeType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="shape"/>
            <xsd:enumeration value="circle"/>
            <xsd:enumeration value="rect"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PathShadeProperties">
        <xsd:sequence>
            <xsd:element name="fillToRect" type="CT_RelativeRect" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="path" type="ST_PathShadeType" use="optional"/>
    </xsd:complexType>
    <xsd:group name="EG_ShadeProperties">
        <xsd:choice>
            <xsd:element name="lin" type="CT_LinearShadeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="path" type="CT_PathShadeProperties" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:simpleType name="ST_TileFlipMode">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="x"/>
            <xsd:enumeration value="y"/>
            <xsd:enumeration value="xy"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_GradientStop">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="pos" type="ST_PositiveFixedPercentage" use="required"/>
```



```
    </xsd:complexType>
    <xsd:complexType name="CT_GradientStopList">
        <xsd:sequence>
            <xsd:element name="gs" type="CT_GradientStop" minOccurs="2" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GradientFillProperties">
        <xsd:sequence>
            <xsd:element name="gsLst" type="CT_GradientStopList" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_ShadeProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tileRect" type="CT_RelativeRect" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="flip" type="ST_TileFlipMode" use="optional"/>
        <xsd:attribute name="rotWithShape" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TileInfoProperties">
        <xsd:attribute name="tx" type="ST_Coordinate" use="optional"/>
        <xsd:attribute name="ty" type="ST_Coordinate" use="optional"/>
        <xsd:attribute name="sx" type="ST_Percentage" use="optional"/>
        <xsd:attribute name="sy" type="ST_Percentage" use="optional"/>
        <xsd:attribute name="flip" type="ST_TileFlipMode" use="optional"/>
        <xsd:attribute name="algn" type="ST_RectAlignment" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_StretchInfoProperties">
        <xsd:sequence>
            <xsd:element name="fillRect" type="CT_RelativeRect" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_FillModeProperties">
        <xsd:choice>
            <xsd:element name="tile" type="CT_TileInfoProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="stretch" type="CT_StretchInfoProperties" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:simpleType name="ST_BlipCompression">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="email"/>
            <xsd:enumeration value="screen"/>
            <xsd:enumeration value="print"/>
            <xsd:enumeration value="hqprint"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Blip">
        <xsd:sequence>
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="alphaBiLevel" type="CT_AlphaBiLevelEffect" minOccurs="1"
                  maxOccurs="1"/>
                <xsd:element name="alphaCeiling" type="CT_AlphaCeilingEffect" minOccurs="1"
                  maxOccurs="1"/>
                <xsd:element name="alphaFloor" type="CT_AlphaFloorEffect" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="alphaInv" type="CT_AlphaInverseEffect" minOccurs="1" maxOccurs="1"/>
```

```
                <xsd:element name="alphaMod" type="CT_AlphaModulateEffect" minOccurs="1"
                  maxOccurs="1"/>
                <xsd:element name="alphaModFix" type="CT_AlphaModulateFixedEffect" minOccurs="1"
                  maxOccurs="1"/>
                <xsd:element name="alphaRepl" type="CT_AlphaReplaceEffect" minOccurs="1"
                  maxOccurs="1"/>
                <xsd:element name="biLevel" type="CT_BiLevelEffect" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="blur" type="CT_BlurEffect" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="clrChange" type="CT_ColorChangeEffect" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="clrRepl" type="CT_ColorReplaceEffect" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="duotone" type="CT_DuotoneEffect" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="fillOverlay" type="CT_FillOverlayEffect" minOccurs="1"
                  maxOccurs="1"/>
                <xsd:element name="grayscl" type="CT_GrayscaleEffect" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="hsl" type="CT_HSLEffect" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="lum" type="CT_LuminanceEffect" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="tint" type="CT_TintEffect" minOccurs="1" maxOccurs="1"/>
            </xsd:choice>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_Blob"/>
        <xsd:attribute name="cstate" type="ST_BlipCompression" use="optional" default="none"/>
    </xsd:complexType>
    <xsd:complexType name="CT_BlipFillProperties">
        <xsd:sequence>
            <xsd:element name="blip" type="CT_Blip" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="srcRect" type="CT_RelativeRect" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_FillModeProperties" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="dpi" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="rotWithShape" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PresetPatternVal">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="pct5"/>
            <xsd:enumeration value="pct10"/>
            <xsd:enumeration value="pct20"/>
            <xsd:enumeration value="pct25"/>
            <xsd:enumeration value="pct30"/>
            <xsd:enumeration value="pct40"/>
            <xsd:enumeration value="pct50"/>
            <xsd:enumeration value="pct60"/>
            <xsd:enumeration value="pct70"/>
            <xsd:enumeration value="pct75"/>
            <xsd:enumeration value="pct80"/>
            <xsd:enumeration value="pct90"/>
            <xsd:enumeration value="horz"/>
            <xsd:enumeration value="vert"/>
            <xsd:enumeration value="ltHorz"/>
            <xsd:enumeration value="ltVert"/>
            <xsd:enumeration value="dkHorz"/>
            <xsd:enumeration value="dkVert"/>
            <xsd:enumeration value="narHorz"/>
```



```
            <xsd:enumeration value="narVert"/>
            <xsd:enumeration value="dashHorz"/>
            <xsd:enumeration value="dashVert"/>
            <xsd:enumeration value="cross"/>
            <xsd:enumeration value="dnDiag"/>
            <xsd:enumeration value="upDiag"/>
            <xsd:enumeration value="ltDnDiag"/>
            <xsd:enumeration value="ltUpDiag"/>
            <xsd:enumeration value="dkDnDiag"/>
            <xsd:enumeration value="dkUpDiag"/>
            <xsd:enumeration value="wdDnDiag"/>
            <xsd:enumeration value="wdUpDiag"/>
            <xsd:enumeration value="dashDnDiag"/>
            <xsd:enumeration value="dashUpDiag"/>
            <xsd:enumeration value="diagCross"/>
            <xsd:enumeration value="smCheck"/>
            <xsd:enumeration value="lgCheck"/>
            <xsd:enumeration value="smGrid"/>
            <xsd:enumeration value="lgGrid"/>
            <xsd:enumeration value="dotGrid"/>
            <xsd:enumeration value="smConfetti"/>
            <xsd:enumeration value="lgConfetti"/>
            <xsd:enumeration value="horzBrick"/>
            <xsd:enumeration value="diagBrick"/>
            <xsd:enumeration value="solidDmnd"/>
            <xsd:enumeration value="openDmnd"/>
            <xsd:enumeration value="dotDmnd"/>
            <xsd:enumeration value="plaid"/>
            <xsd:enumeration value="sphere"/>
            <xsd:enumeration value="weave"/>
            <xsd:enumeration value="divot"/>
            <xsd:enumeration value="shingle"/>
            <xsd:enumeration value="wave"/>
            <xsd:enumeration value="trellis"/>
            <xsd:enumeration value="zigZag"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PatternFillProperties">
        <xsd:sequence>
            <xsd:element name="fgClr" type="CT_Color" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bgClr" type="CT_Color" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="prst" type="ST_PresetPatternVal" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupFillProperties"/>
    <xsd:group name="EG_FillProperties">
        <xsd:choice>
            <xsd:element name="noFill" type="CT_NoFillProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="solidFill" type="CT_SolidColorFillProperties" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="gradFill" type="CT_GradientFillProperties" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="blipFill" type="CT_BlipFillProperties" minOccurs="1" maxOccurs="1"/>
```

```
            <xsd:element name="pattFill" type="CT_PatternFillProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="grpFill" type="CT_GroupFillProperties" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_FillProperties">
        <xsd:sequence>
            <xsd:group ref="EG_FillProperties" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_FillEffect">
        <xsd:sequence>
            <xsd:group ref="EG_FillProperties" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_BlendMode">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="over"/>
            <xsd:enumeration value="mult"/>
            <xsd:enumeration value="screen"/>
            <xsd:enumeration value="darken"/>
            <xsd:enumeration value="lighten"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_FillOverlayEffect">
        <xsd:sequence>
            <xsd:group ref="EG_FillProperties" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="blend" type="ST_BlendMode" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_EffectReference">
        <xsd:attribute name="ref" type="xsd:token" use="required"/>
    </xsd:complexType>
    <xsd:group name="EG_Effect">
        <xsd:choice>
            <xsd:element name="cont" type="CT_EffectContainer" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="effect" type="CT_EffectReference" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="alphaBiLevel" type="CT_AlphaBiLevelEffect" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="alphaCeiling" type="CT_AlphaCeilingEffect" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="alphaFloor" type="CT_AlphaFloorEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="alphaInv" type="CT_AlphaInverseEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="alphaMod" type="CT_AlphaModulateEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="alphaModFix" type="CT_AlphaModulateFixedEffect" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="alphaOutset" type="CT_AlphaOutsetEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="alphaRepl" type="CT_AlphaReplaceEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="biLevel" type="CT_BiLevelEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="blend" type="CT_BlendEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="blur" type="CT_BlurEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="clrChange" type="CT_ColorChangeEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="clrRepl" type="CT_ColorReplaceEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="duotone" type="CT_DuotoneEffect" minOccurs="1" maxOccurs="1"/>
```



```
            <xsd:element name="fill" type="CT_FillEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="fillOverlay" type="CT_FillOverlayEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="glow" type="CT_GlowEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="grayscl" type="CT_GrayscaleEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="hsl" type="CT_HSLEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="innerShdw" type="CT_InnerShadowEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lum" type="CT_LuminanceEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="outerShdw" type="CT_OuterShadowEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="prstShdw" type="CT_PresetShadowEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="reflection" type="CT_ReflectionEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="relOff" type="CT_RelativeOffsetEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="softEdge" type="CT_SoftEdgesEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="tint" type="CT_TintEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="xfrm" type="CT_TransformEffect" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:simpleType name="ST_EffectContainerType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="sib"/>
            <xsd:enumeration value="tree"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_EffectContainer">
        <xsd:group ref="EG_Effect" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:attribute name="type" type="ST_EffectContainerType" use="optional" default="sib"/>
        <xsd:attribute name="name" type="xsd:token" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AlphaModulateEffect">
        <xsd:sequence>
            <xsd:element name="cont" type="CT_EffectContainer" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_BlendEffect">
        <xsd:sequence>
            <xsd:element name="cont" type="CT_EffectContainer" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="blend" type="ST_BlendMode" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_EffectList">
        <xsd:sequence>
            <xsd:element name="blur" type="CT_BlurEffect" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="fillOverlay" type="CT_FillOverlayEffect" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="glow" type="CT_GlowEffect" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="innerShdw" type="CT_InnerShadowEffect" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="outerShdw" type="CT_OuterShadowEffect" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="prstShdw" type="CT_PresetShadowEffect" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="reflection" type="CT_ReflectionEffect" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="softEdge" type="CT_SoftEdgesEffect" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_EffectProperties">
        <xsd:choice>
            <xsd:element name="effectLst" type="CT_EffectList" minOccurs="1" maxOccurs="1"/>
```

```
            <xsd:element name="effectDag" type="CT_EffectContainer" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_EffectProperties">
        <xsd:sequence>
            <xsd:group ref="EG_EffectProperties" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="blip" type="CT_Blip"/>
    <xsd:simpleType name="ST_ShapeType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="line"/>
            <xsd:enumeration value="lineInv"/>
            <xsd:enumeration value="triangle"/>
            <xsd:enumeration value="rtTriangle"/>
            <xsd:enumeration value="rect"/>
            <xsd:enumeration value="diamond"/>
            <xsd:enumeration value="parallelogram"/>
            <xsd:enumeration value="trapezoid"/>
            <xsd:enumeration value="nonIsoscelesTrapezoid"/>
            <xsd:enumeration value="pentagon"/>
            <xsd:enumeration value="hexagon"/>
            <xsd:enumeration value="heptagon"/>
            <xsd:enumeration value="octagon"/>
            <xsd:enumeration value="decagon"/>
            <xsd:enumeration value="dodecagon"/>
            <xsd:enumeration value="star4"/>
            <xsd:enumeration value="star5"/>
            <xsd:enumeration value="star6"/>
            <xsd:enumeration value="star7"/>
            <xsd:enumeration value="star8"/>
            <xsd:enumeration value="star10"/>
            <xsd:enumeration value="star12"/>
            <xsd:enumeration value="star16"/>
            <xsd:enumeration value="star24"/>
            <xsd:enumeration value="star32"/>
            <xsd:enumeration value="roundRect"/>
            <xsd:enumeration value="round1Rect"/>
            <xsd:enumeration value="round2SameRect"/>
            <xsd:enumeration value="round2DiagRect"/>
            <xsd:enumeration value="snipRoundRect"/>
            <xsd:enumeration value="snip1Rect"/>
            <xsd:enumeration value="snip2SameRect"/>
            <xsd:enumeration value="snip2DiagRect"/>
            <xsd:enumeration value="plaque"/>
            <xsd:enumeration value="ellipse"/>
            <xsd:enumeration value="teardrop"/>
            <xsd:enumeration value="homePlate"/>
            <xsd:enumeration value="chevron"/>
            <xsd:enumeration value="pieWedge"/>
            <xsd:enumeration value="pie"/>
            <xsd:enumeration value="blockArc"/>
            <xsd:enumeration value="donut"/>
```



```
            <xsd:enumeration value="noSmoking"/>
            <xsd:enumeration value="rightArrow"/>
            <xsd:enumeration value="leftArrow"/>
            <xsd:enumeration value="upArrow"/>
            <xsd:enumeration value="downArrow"/>
            <xsd:enumeration value="stripedRightArrow"/>
            <xsd:enumeration value="notchedRightArrow"/>
            <xsd:enumeration value="bentUpArrow"/>
            <xsd:enumeration value="leftRightArrow"/>
            <xsd:enumeration value="upDownArrow"/>
            <xsd:enumeration value="leftUpArrow"/>
            <xsd:enumeration value="leftRightUpArrow"/>
            <xsd:enumeration value="quadArrow"/>
            <xsd:enumeration value="leftArrowCallout"/>
            <xsd:enumeration value="rightArrowCallout"/>
            <xsd:enumeration value="upArrowCallout"/>
            <xsd:enumeration value="downArrowCallout"/>
            <xsd:enumeration value="leftRightArrowCallout"/>
            <xsd:enumeration value="upDownArrowCallout"/>
            <xsd:enumeration value="quadArrowCallout"/>
            <xsd:enumeration value="bentArrow"/>
            <xsd:enumeration value="uturnArrow"/>
            <xsd:enumeration value="circularArrow"/>
            <xsd:enumeration value="leftCircularArrow"/>
            <xsd:enumeration value="leftRightCircularArrow"/>
            <xsd:enumeration value="curvedRightArrow"/>
            <xsd:enumeration value="curvedLeftArrow"/>
            <xsd:enumeration value="curvedUpArrow"/>
            <xsd:enumeration value="curvedDownArrow"/>
            <xsd:enumeration value="swooshArrow"/>
            <xsd:enumeration value="cube"/>
            <xsd:enumeration value="can"/>
            <xsd:enumeration value="lightningBolt"/>
            <xsd:enumeration value="heart"/>
            <xsd:enumeration value="sun"/>
            <xsd:enumeration value="moon"/>
            <xsd:enumeration value="smileyFace"/>
            <xsd:enumeration value="irregularSeal1"/>
            <xsd:enumeration value="irregularSeal2"/>
            <xsd:enumeration value="foldedCorner"/>
            <xsd:enumeration value="bevel"/>
            <xsd:enumeration value="frame"/>
            <xsd:enumeration value="halfFrame"/>
            <xsd:enumeration value="corner"/>
            <xsd:enumeration value="diagStripe"/>
            <xsd:enumeration value="chord"/>
            <xsd:enumeration value="arc"/>
            <xsd:enumeration value="leftBracket"/>
            <xsd:enumeration value="rightBracket"/>
            <xsd:enumeration value="leftBrace"/>
            <xsd:enumeration value="rightBrace"/>
            <xsd:enumeration value="bracketPair"/>
            <xsd:enumeration value="bracePair"/>
```

```
            <xsd:enumeration value="straightConnector1"/>
            <xsd:enumeration value="bentConnector2"/>
            <xsd:enumeration value="bentConnector3"/>
            <xsd:enumeration value="bentConnector4"/>
            <xsd:enumeration value="bentConnector5"/>
            <xsd:enumeration value="curvedConnector2"/>
            <xsd:enumeration value="curvedConnector3"/>
            <xsd:enumeration value="curvedConnector4"/>
            <xsd:enumeration value="curvedConnector5"/>
            <xsd:enumeration value="callout1"/>
            <xsd:enumeration value="callout2"/>
            <xsd:enumeration value="callout3"/>
            <xsd:enumeration value="accentCallout1"/>
            <xsd:enumeration value="accentCallout2"/>
            <xsd:enumeration value="accentCallout3"/>
            <xsd:enumeration value="borderCallout1"/>
            <xsd:enumeration value="borderCallout2"/>
            <xsd:enumeration value="borderCallout3"/>
            <xsd:enumeration value="accentBorderCallout1"/>
            <xsd:enumeration value="accentBorderCallout2"/>
            <xsd:enumeration value="accentBorderCallout3"/>
            <xsd:enumeration value="wedgeRectCallout"/>
            <xsd:enumeration value="wedgeRoundRectCallout"/>
            <xsd:enumeration value="wedgeEllipseCallout"/>
            <xsd:enumeration value="cloudCallout"/>
            <xsd:enumeration value="cloud"/>
            <xsd:enumeration value="ribbon"/>
            <xsd:enumeration value="ribbon2"/>
            <xsd:enumeration value="ellipseRibbon"/>
            <xsd:enumeration value="ellipseRibbon2"/>
            <xsd:enumeration value="leftRightRibbon"/>
            <xsd:enumeration value="verticalScroll"/>
            <xsd:enumeration value="horizontalScroll"/>
            <xsd:enumeration value="wave"/>
            <xsd:enumeration value="doubleWave"/>
            <xsd:enumeration value="plus"/>
            <xsd:enumeration value="flowChartProcess"/>
            <xsd:enumeration value="flowChartDecision"/>
            <xsd:enumeration value="flowChartInputOutput"/>
            <xsd:enumeration value="flowChartPredefinedProcess"/>
            <xsd:enumeration value="flowChartInternalStorage"/>
            <xsd:enumeration value="flowChartDocument"/>
            <xsd:enumeration value="flowChartMultidocument"/>
            <xsd:enumeration value="flowChartTerminator"/>
            <xsd:enumeration value="flowChartPreparation"/>
            <xsd:enumeration value="flowChartManualInput"/>
            <xsd:enumeration value="flowChartManualOperation"/>
            <xsd:enumeration value="flowChartConnector"/>
            <xsd:enumeration value="flowChartPunchedCard"/>
            <xsd:enumeration value="flowChartPunchedTape"/>
            <xsd:enumeration value="flowChartSummingJunction"/>
            <xsd:enumeration value="flowChartOr"/>
            <xsd:enumeration value="flowChartCollate"/>
```



```
            <xsd:enumeration value="flowChartSort"/>
            <xsd:enumeration value="flowChartExtract"/>
            <xsd:enumeration value="flowChartMerge"/>
            <xsd:enumeration value="flowChartOfflineStorage"/>
            <xsd:enumeration value="flowChartOnlineStorage"/>
            <xsd:enumeration value="flowChartMagneticTape"/>
            <xsd:enumeration value="flowChartMagneticDisk"/>
            <xsd:enumeration value="flowChartMagneticDrum"/>
            <xsd:enumeration value="flowChartDisplay"/>
            <xsd:enumeration value="flowChartDelay"/>
            <xsd:enumeration value="flowChartAlternateProcess"/>
            <xsd:enumeration value="flowChartOffpageConnector"/>
            <xsd:enumeration value="actionButtonBlank"/>
            <xsd:enumeration value="actionButtonHome"/>
            <xsd:enumeration value="actionButtonHelp"/>
            <xsd:enumeration value="actionButtonInformation"/>
            <xsd:enumeration value="actionButtonForwardNext"/>
            <xsd:enumeration value="actionButtonBackPrevious"/>
            <xsd:enumeration value="actionButtonEnd"/>
            <xsd:enumeration value="actionButtonBeginning"/>
            <xsd:enumeration value="actionButtonReturn"/>
            <xsd:enumeration value="actionButtonDocument"/>
            <xsd:enumeration value="actionButtonSound"/>
            <xsd:enumeration value="actionButtonMovie"/>
            <xsd:enumeration value="gear6"/>
            <xsd:enumeration value="gear9"/>
            <xsd:enumeration value="funnel"/>
            <xsd:enumeration value="mathPlus"/>
            <xsd:enumeration value="mathMinus"/>
            <xsd:enumeration value="mathMultiply"/>
            <xsd:enumeration value="mathDivide"/>
            <xsd:enumeration value="mathEqual"/>
            <xsd:enumeration value="mathNotEqual"/>
            <xsd:enumeration value="cornerTabs"/>
            <xsd:enumeration value="squareTabs"/>
            <xsd:enumeration value="plaqueTabs"/>
            <xsd:enumeration value="chartX"/>
            <xsd:enumeration value="chartStar"/>
            <xsd:enumeration value="chartPlus"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextShapeType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="textNoShape"/>
            <xsd:enumeration value="textPlain"/>
            <xsd:enumeration value="textStop"/>
            <xsd:enumeration value="textTriangle"/>
            <xsd:enumeration value="textTriangleInverted"/>
            <xsd:enumeration value="textChevron"/>
            <xsd:enumeration value="textChevronInverted"/>
            <xsd:enumeration value="textRingInside"/>
            <xsd:enumeration value="textRingOutside"/>
            <xsd:enumeration value="textArchUp"/>
```

```
            <xsd:enumeration value="textArchDown"/>
            <xsd:enumeration value="textCircle"/>
            <xsd:enumeration value="textButton"/>
            <xsd:enumeration value="textArchUpPour"/>
            <xsd:enumeration value="textArchDownPour"/>
            <xsd:enumeration value="textCirclePour"/>
            <xsd:enumeration value="textButtonPour"/>
            <xsd:enumeration value="textCurveUp"/>
            <xsd:enumeration value="textCurveDown"/>
            <xsd:enumeration value="textCanUp"/>
            <xsd:enumeration value="textCanDown"/>
            <xsd:enumeration value="textWave1"/>
            <xsd:enumeration value="textWave2"/>
            <xsd:enumeration value="textDoubleWave1"/>
            <xsd:enumeration value="textWave4"/>
            <xsd:enumeration value="textInflate"/>
            <xsd:enumeration value="textDeflate"/>
            <xsd:enumeration value="textInflateBottom"/>
            <xsd:enumeration value="textDeflateBottom"/>
            <xsd:enumeration value="textInflateTop"/>
            <xsd:enumeration value="textDeflateTop"/>
            <xsd:enumeration value="textDeflateInflate"/>
            <xsd:enumeration value="textDeflateInflateDeflate"/>
            <xsd:enumeration value="textFadeRight"/>
            <xsd:enumeration value="textFadeLeft"/>
            <xsd:enumeration value="textFadeUp"/>
            <xsd:enumeration value="textFadeDown"/>
            <xsd:enumeration value="textSlantUp"/>
            <xsd:enumeration value="textSlantDown"/>
            <xsd:enumeration value="textCascadeUp"/>
            <xsd:enumeration value="textCascadeDown"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_GeomGuideName">
        <xsd:restriction base="xsd:token"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_GeomGuideFormula">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_GeomGuide">
        <xsd:attribute name="name" type="ST_GeomGuideName" use="required"/>
        <xsd:attribute name="fmla" type="ST_GeomGuideFormula" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GeomGuideList">
        <xsd:sequence>
            <xsd:element name="gd" type="CT_GeomGuide" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_AdjCoordinate">
        <xsd:union memberTypes="ST_Coordinate ST_GeomGuideName"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_AdjAngle">
        <xsd:union memberTypes="ST_Angle ST_GeomGuideName"/>
```



```
    </xsd:simpleType>
    <xsd:complexType name="CT_AdjPoint2D">
        <xsd:attribute name="x" type="ST_AdjCoordinate" use="required"/>
        <xsd:attribute name="y" type="ST_AdjCoordinate" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GeomRect">
        <xsd:attribute name="l" type="ST_AdjCoordinate" use="required"/>
        <xsd:attribute name="t" type="ST_AdjCoordinate" use="required"/>
        <xsd:attribute name="r" type="ST_AdjCoordinate" use="required"/>
        <xsd:attribute name="b" type="ST_AdjCoordinate" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_XYAdjustHandle">
        <xsd:sequence>
            <xsd:element name="pos" type="CT_AdjPoint2D" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="gdRefX" type="ST_GeomGuideName" use="optional"/>
        <xsd:attribute name="minX" type="ST_AdjCoordinate" use="optional"/>
        <xsd:attribute name="maxX" type="ST_AdjCoordinate" use="optional"/>
        <xsd:attribute name="gdRefY" type="ST_GeomGuideName" use="optional"/>
        <xsd:attribute name="minY" type="ST_AdjCoordinate" use="optional"/>
        <xsd:attribute name="maxY" type="ST_AdjCoordinate" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PolarAdjustHandle">
        <xsd:sequence>
            <xsd:element name="pos" type="CT_AdjPoint2D" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="gdRefR" type="ST_GeomGuideName" use="optional"/>
        <xsd:attribute name="minR" type="ST_AdjCoordinate" use="optional"/>
        <xsd:attribute name="maxR" type="ST_AdjCoordinate" use="optional"/>
        <xsd:attribute name="gdRefAng" type="ST_GeomGuideName" use="optional"/>
        <xsd:attribute name="minAng" type="ST_AdjAngle" use="optional"/>
        <xsd:attribute name="maxAng" type="ST_AdjAngle" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ConnectionSite">
        <xsd:sequence>
            <xsd:element name="pos" type="CT_AdjPoint2D" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="ang" type="ST_AdjAngle" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AdjustHandleList">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="ahXY" type="CT_XYAdjustHandle" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="ahPolar" type="CT_PolarAdjustHandle" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_ConnectionSiteList">
        <xsd:sequence>
            <xsd:element name="cxn" type="CT_ConnectionSite" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Connection">
        <xsd:attribute name="id" type="ST_DrawingElementId" use="required"/>
        <xsd:attribute name="idx" type="xsd:unsignedInt" use="required"/>
```

```
    </xsd:complexType>
    <xsd:complexType name="CT_Path2DMoveTo">
        <xsd:sequence>
            <xsd:element name="pt" type="CT_AdjPoint2D" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Path2DLineTo">
        <xsd:sequence>
            <xsd:element name="pt" type="CT_AdjPoint2D" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Path2DArcTo">
        <xsd:attribute name="wR" type="ST_AdjCoordinate" use="required"/>
        <xsd:attribute name="hR" type="ST_AdjCoordinate" use="required"/>
        <xsd:attribute name="stAng" type="ST_AdjAngle" use="required"/>
        <xsd:attribute name="swAng" type="ST_AdjAngle" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Path2DQuadBezierTo">
        <xsd:sequence>
            <xsd:element name="pt" type="CT_AdjPoint2D" minOccurs="2" maxOccurs="2"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Path2DCubicBezierTo">
        <xsd:sequence>
            <xsd:element name="pt" type="CT_AdjPoint2D" minOccurs="3" maxOccurs="3"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Path2DClose"/>
    <xsd:simpleType name="ST_PathFillMode">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="norm"/>
            <xsd:enumeration value="lighten"/>
            <xsd:enumeration value="lightenLess"/>
            <xsd:enumeration value="darken"/>
            <xsd:enumeration value="darkenLess"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Path2D">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="close" type="CT_Path2DClose" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="moveTo" type="CT_Path2DMoveTo" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lnTo" type="CT_Path2DLineTo" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="arcTo" type="CT_Path2DArcTo" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="quadBezTo" type="CT_Path2DQuadBezierTo" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cubicBezTo" type="CT_Path2DCubicBezierTo" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
        <xsd:attribute name="w" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="h" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="fill" type="ST_PathFillMode" use="optional" default="norm"/>
        <xsd:attribute name="stroke" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="extrusionOk" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
```



```
    <xsd:complexType name="CT_Path2DList">
        <xsd:sequence>
            <xsd:element name="path" type="CT_Path2D" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_PresetGeometry2D">
        <xsd:sequence>
            <xsd:element name="avLst" type="CT_GeomGuideList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="prst" type="ST_ShapeType" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PresetTextShape">
        <xsd:sequence>
            <xsd:element name="avLst" type="CT_GeomGuideList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="prst" type="ST_TextShapeType" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomGeometry2D">
        <xsd:sequence>
            <xsd:element name="avLst" type="CT_GeomGuideList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="gdLst" type="CT_GeomGuideList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ahLst" type="CT_AdjustHandleList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cxnLst" type="CT_ConnectionSiteList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="rect" type="CT_GeomRect" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pathLst" type="CT_Path2DList" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_Geometry">
        <xsd:choice>
            <xsd:element name="custGeom" type="CT_CustomGeometry2D" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="prstGeom" type="CT_PresetGeometry2D" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:group name="EG_TextGeometry">
        <xsd:choice>
            <xsd:element name="custGeom" type="CT_CustomGeometry2D" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="prstTxWarp" type="CT_PresetTextShape" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:simpleType name="ST_LineEndType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="triangle"/>
            <xsd:enumeration value="stealth"/>
            <xsd:enumeration value="diamond"/>
            <xsd:enumeration value="oval"/>
            <xsd:enumeration value="arrow"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_LineEndWidth">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="sm"/>
            <xsd:enumeration value="med"/>
```

```
            <xsd:enumeration value="lg"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_LineEndLength">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="sm"/>
            <xsd:enumeration value="med"/>
            <xsd:enumeration value="lg"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_LineEndProperties">
        <xsd:attribute name="type" type="ST_LineEndType" use="optional"/>
        <xsd:attribute name="w" type="ST_LineEndWidth" use="optional"/>
        <xsd:attribute name="len" type="ST_LineEndLength" use="optional"/>
    </xsd:complexType>
    <xsd:group name="EG_LineFillProperties">
        <xsd:choice>
            <xsd:element name="noFill" type="CT_NoFillProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="solidFill" type="CT_SolidColorFillProperties" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="gradFill" type="CT_GradientFillProperties" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="pattFill" type="CT_PatternFillProperties" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_LineJoinBevel"/>
    <xsd:complexType name="CT_LineJoinRound"/>
    <xsd:complexType name="CT_LineJoinMiterProperties">
        <xsd:attribute name="lim" type="ST_PositivePercentage" use="optional"/>
    </xsd:complexType>
    <xsd:group name="EG_LineJoinProperties">
        <xsd:choice>
            <xsd:element name="round" type="CT_LineJoinRound" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="bevel" type="CT_LineJoinBevel" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="miter" type="CT_LineJoinMiterProperties" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:simpleType name="ST_PresetLineDashVal">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="solid"/>
            <xsd:enumeration value="dot"/>
            <xsd:enumeration value="dash"/>
            <xsd:enumeration value="lgDash"/>
            <xsd:enumeration value="dashDot"/>
            <xsd:enumeration value="lgDashDot"/>
            <xsd:enumeration value="lgDashDotDot"/>
            <xsd:enumeration value="sysDash"/>
            <xsd:enumeration value="sysDot"/>
            <xsd:enumeration value="sysDashDot"/>
            <xsd:enumeration value="sysDashDotDot"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PresetLineDashProperties">
```



```
        <xsd:attribute name="val" type="ST_PresetLineDashVal" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DashStop">
        <xsd:attribute name="d" type="ST_PositivePercentage" use="required"/>
        <xsd:attribute name="sp" type="ST_PositivePercentage" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DashStopList">
        <xsd:sequence>
            <xsd:element name="ds" type="CT_DashStop" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_LineDashProperties">
        <xsd:choice>
            <xsd:element name="prstDash" type="CT_PresetLineDashProperties" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="custDash" type="CT_DashStopList" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:simpleType name="ST_LineCap">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="rnd"/>
            <xsd:enumeration value="sq"/>
            <xsd:enumeration value="flat"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_LineWidth">
        <xsd:restriction base="ST_Coordinate32Unqualified">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="20116800"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PenAlignment">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="ctr"/>
            <xsd:enumeration value="in"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_CompoundLine">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="sng"/>
            <xsd:enumeration value="dbl"/>
            <xsd:enumeration value="thickThin"/>
            <xsd:enumeration value="thinThick"/>
            <xsd:enumeration value="tri"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_LineProperties">
        <xsd:sequence>
            <xsd:group ref="EG_LineFillProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_LineDashProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_LineJoinProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="headEnd" type="CT_LineEndProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tailEnd" type="CT_LineEndProperties" minOccurs="0" maxOccurs="1"/>
```

```
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="w" type="ST_LineWidth" use="optional"/>
        <xsd:attribute name="cap" type="ST_LineCap" use="optional"/>
        <xsd:attribute name="cmpd" type="ST_CompoundLine" use="optional"/>
        <xsd:attribute name="algn" type="ST_PenAlignment" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_ShapeID">
        <xsd:restriction base="xsd:token"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_ShapeProperties">
        <xsd:sequence>
            <xsd:element name="xfrm" type="CT_Transform2D" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_Geometry" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_FillProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ln" type="CT_LineProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_EffectProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="scene3d" type="CT_Scene3D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sp3d" type="CT_Shape3D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="bwMode" type="ST_BlackWhiteMode" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupShapeProperties">
        <xsd:sequence>
            <xsd:element name="xfrm" type="CT_GroupTransform2D" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_FillProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_EffectProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="scene3d" type="CT_Scene3D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="bwMode" type="ST_BlackWhiteMode" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_StyleMatrixReference">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="idx" type="ST_StyleMatrixColumnIndex" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FontReference">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="idx" type="ST_FontCollectionIndex" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ShapeStyle">
        <xsd:sequence>
            <xsd:element name="lnRef" type="CT_StyleMatrixReference" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="fillRef" type="CT_StyleMatrixReference" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="effectRef" type="CT_StyleMatrixReference" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="fontRef" type="CT_FontReference" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
```



```
    <xsd:complexType name="CT_DefaultShapeDefinition">
        <xsd:sequence>
            <xsd:element name="spPr" type="CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="bodyPr" type="CT_TextBodyProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lstStyle" type="CT_TextListStyle" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="style" type="CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ObjectStyleDefaults">
        <xsd:sequence>
            <xsd:element name="spDef" type="CT_DefaultShapeDefinition" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="lnDef" type="CT_DefaultShapeDefinition" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="txDef" type="CT_DefaultShapeDefinition" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_EmptyElement"/>
    <xsd:complexType name="CT_ColorMapping">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="bg1" type="ST_ColorSchemeIndex" use="required"/>
        <xsd:attribute name="tx1" type="ST_ColorSchemeIndex" use="required"/>
        <xsd:attribute name="bg2" type="ST_ColorSchemeIndex" use="required"/>
        <xsd:attribute name="tx2" type="ST_ColorSchemeIndex" use="required"/>
        <xsd:attribute name="accent1" type="ST_ColorSchemeIndex" use="required"/>
        <xsd:attribute name="accent2" type="ST_ColorSchemeIndex" use="required"/>
        <xsd:attribute name="accent3" type="ST_ColorSchemeIndex" use="required"/>
        <xsd:attribute name="accent4" type="ST_ColorSchemeIndex" use="required"/>
        <xsd:attribute name="accent5" type="ST_ColorSchemeIndex" use="required"/>
        <xsd:attribute name="accent6" type="ST_ColorSchemeIndex" use="required"/>
        <xsd:attribute name="hlink" type="ST_ColorSchemeIndex" use="required"/>
        <xsd:attribute name="folHlink" type="ST_ColorSchemeIndex" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ColorMappingOverride">
        <xsd:sequence>
            <xsd:choice minOccurs="1" maxOccurs="1">
                <xsd:element name="masterClrMapping" type="CT_EmptyElement"/>
                <xsd:element name="overrideClrMapping" type="CT_ColorMapping"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ColorSchemeAndMapping">
        <xsd:sequence>
            <xsd:element name="clrScheme" type="CT_ColorScheme" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="clrMap" type="CT_ColorMapping" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ColorSchemeList">
        <xsd:sequence>
            <xsd:element name="extraClrScheme" type="CT_ColorSchemeAndMapping" minOccurs="0"
              maxOccurs="unbounded"/>
```

```
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_OfficeStyleSheet">
        <xsd:sequence>
            <xsd:element name="themeElements" type="CT_BaseStyles" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="objectDefaults" type="CT_ObjectStyleDefaults" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="extraClrSchemeLst" type="CT_ColorSchemeList" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="custClrLst" type="CT_CustomColorList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="optional" default=""/>
    </xsd:complexType>
    <xsd:complexType name="CT_BaseStylesOverride">
        <xsd:sequence>
            <xsd:element name="clrScheme" type="CT_ColorScheme" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="fontScheme" type="CT_FontScheme" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="fmtScheme" type="CT_StyleMatrix" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ClipboardStyleSheet">
        <xsd:sequence>
            <xsd:element name="themeElements" type="CT_BaseStyles" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="clrMap" type="CT_ColorMapping" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="theme" type="CT_OfficeStyleSheet"/>
    <xsd:element name="themeOverride" type="CT_BaseStylesOverride"/>
    <xsd:element name="themeManager" type="CT_EmptyElement"/>
    <xsd:complexType name="CT_TableCellProperties">
        <xsd:sequence>
            <xsd:element name="lnL" type="CT_LineProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="lnR" type="CT_LineProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="lnT" type="CT_LineProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="lnB" type="CT_LineProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="lnTlToBr" type="CT_LineProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="lnBlToTr" type="CT_LineProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cell3D" type="CT_Cell3D" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_FillProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="headers" type="CT_Headers" minOccurs="0"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="marL" type="ST_Coordinate32" use="optional" default="91440"/>
        <xsd:attribute name="marR" type="ST_Coordinate32" use="optional" default="91440"/>
        <xsd:attribute name="marT" type="ST_Coordinate32" use="optional" default="45720"/>
        <xsd:attribute name="marB" type="ST_Coordinate32" use="optional" default="45720"/>
        <xsd:attribute name="vert" type="ST_TextVerticalType" use="optional" default="horz"/>
        <xsd:attribute name="anchor" type="ST_TextAnchoringType" use="optional" default="t"/>
        <xsd:attribute name="anchorCtr" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="horzOverflow" type="ST_TextHorzOverflowType" use="optional"
          default="clip"/>
    </xsd:complexType>
```



```
    <xsd:complexType name="CT_Headers">
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="header" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TableCol">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="w" type="ST_Coordinate" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TableGrid">
        <xsd:sequence>
            <xsd:element name="gridCol" type="CT_TableCol" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TableCell">
        <xsd:sequence>
            <xsd:element name="txBody" type="CT_TextBody" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tcPr" type="CT_TableCellProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="rowSpan" type="xsd:int" use="optional" default="1"/>
        <xsd:attribute name="gridSpan" type="xsd:int" use="optional" default="1"/>
        <xsd:attribute name="hMerge" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="vMerge" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="id" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TableRow">
        <xsd:sequence>
            <xsd:element name="tc" type="CT_TableCell" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="h" type="ST_Coordinate" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TableProperties">
        <xsd:sequence>
            <xsd:group ref="EG_FillProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_EffectProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:choice minOccurs="0" maxOccurs="1">
                <xsd:element name="tableStyle" type="CT_TableStyle"/>
                <xsd:element name="tableStyleId" type="s:ST_Guid"/>
            </xsd:choice>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="rtl" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="firstRow" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="firstCol" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="lastRow" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="lastCol" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="bandRow" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="bandCol" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
```

```
    <xsd:complexType name="CT_Table">
        <xsd:sequence>
            <xsd:element name="tblPr" type="CT_TableProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblGrid" type="CT_TableGrid" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="tr" type="CT_TableRow" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="tbl" type="CT_Table"/>
    <xsd:complexType name="CT_Cell3D">
        <xsd:sequence>
            <xsd:element name="bevel" type="CT_Bevel" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lightRig" type="CT_LightRig" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="prstMaterial" type="ST_PresetMaterialType" use="optional"
          default="plastic"/>
    </xsd:complexType>
    <xsd:group name="EG_ThemeableFillStyle">
        <xsd:choice>
            <xsd:element name="fill" type="CT_FillProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="fillRef" type="CT_StyleMatrixReference" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_ThemeableLineStyle">
        <xsd:choice>
            <xsd:element name="ln" type="CT_LineProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lnRef" type="CT_StyleMatrixReference" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:group name="EG_ThemeableEffectStyle">
        <xsd:choice>
            <xsd:element name="effect" type="CT_EffectProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="effectRef" type="CT_StyleMatrixReference" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:group name="EG_ThemeableFontStyles">
        <xsd:choice>
            <xsd:element name="font" type="CT_FontCollection" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="fontRef" type="CT_FontReference" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:simpleType name="ST_OnOffStyleType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="on"/>
            <xsd:enumeration value="off"/>
            <xsd:enumeration value="def"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TableStyleTextStyle">
        <xsd:sequence>
            <xsd:group ref="EG_ThemeableFontStyles" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_ColorChoice" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
```



```
        </xsd:sequence>
        <xsd:attribute name="b" type="ST_OnOffStyleType" use="optional" default="def"/>
        <xsd:attribute name="i" type="ST_OnOffStyleType" use="optional" default="def"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TableCellBorderStyle">
        <xsd:sequence>
            <xsd:element name="left" type="CT_ThemeableLineStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="right" type="CT_ThemeableLineStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="top" type="CT_ThemeableLineStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bottom" type="CT_ThemeableLineStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="insideH" type="CT_ThemeableLineStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="insideV" type="CT_ThemeableLineStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tl2br" type="CT_ThemeableLineStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tr2bl" type="CT_ThemeableLineStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TableBackgroundStyle">
        <xsd:sequence>
            <xsd:group ref="EG_ThemeableFillStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_ThemeableEffectStyle" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TableStyleCellStyle">
        <xsd:sequence>
            <xsd:element name="tcBdr" type="CT_TableCellBorderStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_ThemeableFillStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cell3D" type="CT_Cell3D" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TablePartStyle">
        <xsd:sequence>
            <xsd:element name="tcTxStyle" type="CT_TableStyleTextStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tcStyle" type="CT_TableStyleCellStyle" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TableStyle">
        <xsd:sequence>
            <xsd:element name="tblBg" type="CT_TableBackgroundStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="wholeTbl" type="CT_TablePartStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="band1H" type="CT_TablePartStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="band2H" type="CT_TablePartStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="band1V" type="CT_TablePartStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="band2V" type="CT_TablePartStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="lastCol" type="CT_TablePartStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="firstCol" type="CT_TablePartStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="lastRow" type="CT_TablePartStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="seCell" type="CT_TablePartStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="swCell" type="CT_TablePartStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="firstRow" type="CT_TablePartStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="neCell" type="CT_TablePartStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="nwCell" type="CT_TablePartStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
```

```
        </xsd:sequence>
        <xsd:attribute name="styleId" type="s:ST_Guid" use="required"/>
        <xsd:attribute name="styleName" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TableStyleList">
        <xsd:sequence>
            <xsd:element name="tblStyle" type="CT_TableStyle" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="def" type="s:ST_Guid" use="required"/>
    </xsd:complexType>
    <xsd:element name="tblStyleLst" type="CT_TableStyleList"/>
    <xsd:complexType name="CT_TextParagraph">
        <xsd:sequence>
            <xsd:element name="pPr" type="CT_TextParagraphProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_TextRun" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="endParaRPr" type="CT_TextCharacterProperties" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_TextAnchoringType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="t"/>
            <xsd:enumeration value="ctr"/>
            <xsd:enumeration value="b"/>
            <xsd:enumeration value="just"/>
            <xsd:enumeration value="dist"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextVertOverflowType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="overflow"/>
            <xsd:enumeration value="ellipsis"/>
            <xsd:enumeration value="clip"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextHorzOverflowType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="overflow"/>
            <xsd:enumeration value="clip"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextVerticalType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="horz"/>
            <xsd:enumeration value="vert"/>
            <xsd:enumeration value="vert270"/>
            <xsd:enumeration value="wordArtVert"/>
            <xsd:enumeration value="eaVert"/>
            <xsd:enumeration value="mongolianVert"/>
            <xsd:enumeration value="wordArtVertRtl"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextWrappingType">
```



```
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="square"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextColumnCount">
        <xsd:restriction base="xsd:int">
            <xsd:minInclusive value="1"/>
            <xsd:maxInclusive value="16"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextListStyle">
        <xsd:sequence>
            <xsd:element name="defPPr" type="CT_TextParagraphProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="lvl1pPr" type="CT_TextParagraphProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="lvl2pPr" type="CT_TextParagraphProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="lvl3pPr" type="CT_TextParagraphProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="lvl4pPr" type="CT_TextParagraphProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="lvl5pPr" type="CT_TextParagraphProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="lvl6pPr" type="CT_TextParagraphProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="lvl7pPr" type="CT_TextParagraphProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="lvl8pPr" type="CT_TextParagraphProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="lvl9pPr" type="CT_TextParagraphProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_TextFontScalePercentOrPercentString">
        <xsd:union memberTypes="ST_TextFontScalePercent s:ST_Percentage"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextFontScalePercent">
        <xsd:restriction base="ST_PercentageDecimal">
            <xsd:minInclusive value="1000"/>
            <xsd:maxInclusive value="100000"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextNormalAutofit">
        <xsd:attribute name="fontScale" type="ST_TextFontScalePercentOrPercentString" use="optional"
          default="100%"/>
        <xsd:attribute name="lnSpcReduction" type="ST_TextSpacingPercentOrPercentString"
          use="optional" default="0%"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TextShapeAutofit"/>
    <xsd:complexType name="CT_TextNoAutofit"/>
    <xsd:group name="EG_TextAutofit">
```

```
        <xsd:choice>
            <xsd:element name="noAutofit" type="CT_TextNoAutofit"/>
            <xsd:element name="normAutofit" type="CT_TextNormalAutofit"/>
            <xsd:element name="spAutoFit" type="CT_TextShapeAutofit"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_TextBodyProperties">
        <xsd:sequence>
            <xsd:element name="prstTxWarp" type="CT_PresetTextShape" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_TextAutofit" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="scene3d" type="CT_Scene3D" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_Text3D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="rot" type="ST_Angle" use="optional"/>
        <xsd:attribute name="spcFirstLastPara" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="vertOverflow" type="ST_TextVertOverflowType" use="optional"/>
        <xsd:attribute name="horzOverflow" type="ST_TextHorzOverflowType" use="optional"/>
        <xsd:attribute name="vert" type="ST_TextVerticalType" use="optional"/>
        <xsd:attribute name="wrap" type="ST_TextWrappingType" use="optional"/>
        <xsd:attribute name="lIns" type="ST_Coordinate32" use="optional"/>
        <xsd:attribute name="tIns" type="ST_Coordinate32" use="optional"/>
        <xsd:attribute name="rIns" type="ST_Coordinate32" use="optional"/>
        <xsd:attribute name="bIns" type="ST_Coordinate32" use="optional"/>
        <xsd:attribute name="numCol" type="ST_TextColumnCount" use="optional"/>
        <xsd:attribute name="spcCol" type="ST_PositiveCoordinate32" use="optional"/>
        <xsd:attribute name="rtlCol" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="fromWordArt" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="anchor" type="ST_TextAnchoringType" use="optional"/>
        <xsd:attribute name="anchorCtr" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="forceAA" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="upright" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="compatLnSpc" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TextBody">
        <xsd:sequence>
            <xsd:element name="bodyPr" type="CT_TextBodyProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lstStyle" type="CT_TextListStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="p" type="CT_TextParagraph" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_TextBulletStartAtNum">
        <xsd:restriction base="xsd:int">
            <xsd:minInclusive value="1"/>
            <xsd:maxInclusive value="32767"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextAutonumberScheme">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="alphaLcParenBoth"/>
            <xsd:enumeration value="alphaUcParenBoth"/>
            <xsd:enumeration value="alphaLcParenR"/>
            <xsd:enumeration value="alphaUcParenR"/>
```



```
            <xsd:enumeration value="alphaLcPeriod"/>
            <xsd:enumeration value="alphaUcPeriod"/>
            <xsd:enumeration value="arabicParenBoth"/>
            <xsd:enumeration value="arabicParenR"/>
            <xsd:enumeration value="arabicPeriod"/>
            <xsd:enumeration value="arabicPlain"/>
            <xsd:enumeration value="romanLcParenBoth"/>
            <xsd:enumeration value="romanUcParenBoth"/>
            <xsd:enumeration value="romanLcParenR"/>
            <xsd:enumeration value="romanUcParenR"/>
            <xsd:enumeration value="romanLcPeriod"/>
            <xsd:enumeration value="romanUcPeriod"/>
            <xsd:enumeration value="circleNumDbPlain"/>
            <xsd:enumeration value="circleNumWdBlackPlain"/>
            <xsd:enumeration value="circleNumWdWhitePlain"/>
            <xsd:enumeration value="arabicDbPeriod"/>
            <xsd:enumeration value="arabicDbPlain"/>
            <xsd:enumeration value="ea1ChsPeriod"/>
            <xsd:enumeration value="ea1ChsPlain"/>
            <xsd:enumeration value="ea1ChtPeriod"/>
            <xsd:enumeration value="ea1ChtPlain"/>
            <xsd:enumeration value="ea1JpnChsDbPeriod"/>
            <xsd:enumeration value="ea1JpnKorPlain"/>
            <xsd:enumeration value="ea1JpnKorPeriod"/>
            <xsd:enumeration value="arabic1Minus"/>
            <xsd:enumeration value="arabic2Minus"/>
            <xsd:enumeration value="hebrew2Minus"/>
            <xsd:enumeration value="thaiAlphaPeriod"/>
            <xsd:enumeration value="thaiAlphaParenR"/>
            <xsd:enumeration value="thaiAlphaParenBoth"/>
            <xsd:enumeration value="thaiNumPeriod"/>
            <xsd:enumeration value="thaiNumParenR"/>
            <xsd:enumeration value="thaiNumParenBoth"/>
            <xsd:enumeration value="hindiAlphaPeriod"/>
            <xsd:enumeration value="hindiNumPeriod"/>
            <xsd:enumeration value="hindiNumParenR"/>
            <xsd:enumeration value="hindiAlpha1Period"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextBulletColorFollowText"/>
    <xsd:group name="EG_TextBulletColor">
        <xsd:choice>
            <xsd:element name="buClrTx" type="CT_TextBulletColorFollowText" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="buClr" type="CT_Color" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:simpleType name="ST_TextBulletSize">
        <xsd:union memberTypes="ST_TextBulletSizePercent ST_TextBulletSizeDecimal"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextBulletSizePercent">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="0*((2[5-9])|([3-9][0-9])|([1-3][0-9][0-9])|400)%"/>
```

```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextBulletSizeDecimal">
        <xsd:restriction base="ST_PercentageDecimal">
            <xsd:minInclusive value="25000"/>
            <xsd:maxInclusive value="400000"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextBulletSizeFollowText"/>
    <xsd:complexType name="CT_TextBulletSizePercent">
        <xsd:attribute name="val" type="ST_TextBulletSizePercent" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TextBulletSizePoint">
        <xsd:attribute name="val" type="ST_TextFontSize" use="required"/>
    </xsd:complexType>
    <xsd:group name="EG_TextBulletSize">
        <xsd:choice>
            <xsd:element name="buSzTx" type="CT_TextBulletSizeFollowText"/>
            <xsd:element name="buSzPct" type="CT_TextBulletSizePercent"/>
            <xsd:element name="buSzPts" type="CT_TextBulletSizePoint"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_TextBulletTypefaceFollowText"/>
    <xsd:group name="EG_TextBulletTypeface">
        <xsd:choice>
            <xsd:element name="buFontTx" type="CT_TextBulletTypefaceFollowText"/>
            <xsd:element name="buFont" type="CT_TextFont"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_TextAutonumberBullet">
        <xsd:attribute name="type" type="ST_TextAutonumberScheme" use="required"/>
        <xsd:attribute name="startAt" type="ST_TextBulletStartAtNum" use="optional" default="1"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TextCharBullet">
        <xsd:attribute name="char" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TextBlipBullet">
        <xsd:sequence>
            <xsd:element name="blip" type="CT_Blip" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TextNoBullet"/>
    <xsd:group name="EG_TextBullet">
        <xsd:choice>
            <xsd:element name="buNone" type="CT_TextNoBullet"/>
            <xsd:element name="buAutoNum" type="CT_TextAutonumberBullet"/>
            <xsd:element name="buChar" type="CT_TextCharBullet"/>
            <xsd:element name="buBlip" type="CT_TextBlipBullet"/>
        </xsd:choice>
    </xsd:group>
    <xsd:simpleType name="ST_TextPoint">
        <xsd:union memberTypes="ST_TextPointUnqualified s:ST_UniversalMeasure"/>
    </xsd:simpleType>
```



```
    <xsd:simpleType name="ST_TextPointUnqualified">
        <xsd:restriction base="xsd:int">
            <xsd:minInclusive value="-400000"/>
            <xsd:maxInclusive value="400000"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextNonNegativePoint">
        <xsd:restriction base="xsd:int">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="400000"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextFontSize">
        <xsd:restriction base="xsd:int">
            <xsd:minInclusive value="100"/>
            <xsd:maxInclusive value="400000"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextTypeface">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PitchFamily">
        <xsd:restriction base="xsd:byte">
        <xsd:enumeration value="00"/>
        <xsd:enumeration value="01"/>
        <xsd:enumeration value="02"/>
        <xsd:enumeration value="16"/>
        <xsd:enumeration value="17"/>
        <xsd:enumeration value="18"/>
        <xsd:enumeration value="32"/>
        <xsd:enumeration value="33"/>
        <xsd:enumeration value="34"/>
        <xsd:enumeration value="48"/>
        <xsd:enumeration value="49"/>
        <xsd:enumeration value="50"/>
        <xsd:enumeration value="64"/>
        <xsd:enumeration value="65"/>
        <xsd:enumeration value="66"/>
        <xsd:enumeration value="80"/>
        <xsd:enumeration value="81"/>
        <xsd:enumeration value="82"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextFont">
        <xsd:attribute name="typeface" type="ST_TextTypeface" use="required"/>
        <xsd:attribute name="panose" type="s:ST_Panose" use="optional"/>
        <xsd:attribute name="pitchFamily" type="ST_PitchFamily" use="optional" default="0"/>
        <xsd:attribute name="charset" type="xsd:byte" use="optional" default="1"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TextUnderlineType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="words"/>
```



```
            <xsd:enumeration value="sng"/>
            <xsd:enumeration value="dbl"/>
            <xsd:enumeration value="heavy"/>
            <xsd:enumeration value="dotted"/>
            <xsd:enumeration value="dottedHeavy"/>
            <xsd:enumeration value="dash"/>
            <xsd:enumeration value="dashHeavy"/>
            <xsd:enumeration value="dashLong"/>
            <xsd:enumeration value="dashLongHeavy"/>
            <xsd:enumeration value="dotDash"/>
            <xsd:enumeration value="dotDashHeavy"/>
            <xsd:enumeration value="dotDotDash"/>
            <xsd:enumeration value="dotDotDashHeavy"/>
            <xsd:enumeration value="wavy"/>
            <xsd:enumeration value="wavyHeavy"/>
            <xsd:enumeration value="wavyDbl"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextUnderlineLineFollowText"/>
    <xsd:complexType name="CT_TextUnderlineFillFollowText"/>
    <xsd:complexType name="CT_TextUnderlineFillGroupWrapper">
        <xsd:group ref="EG_FillProperties" minOccurs="1" maxOccurs="1"/>
    </xsd:complexType>
    <xsd:group name="EG_TextUnderlineLine">
        <xsd:choice>
            <xsd:element name="uLnTx" type="CT_TextUnderlineLineFollowText"/>
            <xsd:element name="uLn" type="CT_LineProperties" minOccurs="0" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:group name="EG_TextUnderlineFill">
        <xsd:choice>
            <xsd:element name="uFillTx" type="CT_TextUnderlineFillFollowText"/>
            <xsd:element name="uFill" type="CT_TextUnderlineFillGroupWrapper"/>
        </xsd:choice>
    </xsd:group>
    <xsd:simpleType name="ST_TextStrikeType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="noStrike"/>
            <xsd:enumeration value="sngStrike"/>
            <xsd:enumeration value="dblStrike"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextCapsType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="small"/>
            <xsd:enumeration value="all"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextCharacterProperties">
        <xsd:sequence>
            <xsd:element name="ln" type="CT_LineProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_FillProperties" minOccurs="0" maxOccurs="1"/>
```



```
            <xsd:group ref="EG_EffectProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="highlight" type="CT_Color" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_TextUnderlineLine" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_TextUnderlineFill" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="latin" type="CT_TextFont" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ea" type="CT_TextFont" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cs" type="CT_TextFont" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sym" type="CT_TextFont" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="hlinkClick" type="CT_Hyperlink" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="hlinkMouseOver" type="CT_Hyperlink" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="rtl" type="CT_Boolean" minOccurs="0"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="kumimoji" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="lang" type="s:ST_Lang" use="optional"/>
        <xsd:attribute name="altLang" type="s:ST_Lang" use="optional"/>
        <xsd:attribute name="sz" type="ST_TextFontSize" use="optional"/>
        <xsd:attribute name="b" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="i" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="u" type="ST_TextUnderlineType" use="optional"/>
        <xsd:attribute name="strike" type="ST_TextStrikeType" use="optional"/>
        <xsd:attribute name="kern" type="ST_TextNonNegativePoint" use="optional"/>
        <xsd:attribute name="cap" type="ST_TextCapsType" use="optional"/>
        <xsd:attribute name="spc" type="ST_TextPoint" use="optional"/>
        <xsd:attribute name="normalizeH" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="baseline" type="ST_Percentage" use="optional"/>
        <xsd:attribute name="noProof" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="dirty" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="err" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="smtClean" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="smtId" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="bmk" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Boolean">
        <xsd:attribute name="val" type="s:ST_OnOff" default="0"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TextSpacingPoint">
        <xsd:restriction base="xsd:int">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="158400"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextSpacingPercentOrPercentString">
        <xsd:union memberTypes="ST_TextSpacingPercent s:ST_Percentage"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextSpacingPercent">
        <xsd:restriction base="ST_PercentageDecimal">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="13200000"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextSpacingPercent">
        <xsd:attribute name="val" type="ST_TextSpacingPercentOrPercentString" use="required"/>
```

```
    </xsd:complexType>
    <xsd:complexType name="CT_TextSpacingPoint">
        <xsd:attribute name="val" type="ST_TextSpacingPoint" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TextMargin">
        <xsd:restriction base="ST_Coordinate32Unqualified">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="51206400"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextIndent">
        <xsd:restriction base="ST_Coordinate32Unqualified">
            <xsd:minInclusive value="-51206400"/>
            <xsd:maxInclusive value="51206400"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextTabAlignType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="l"/>
            <xsd:enumeration value="ctr"/>
            <xsd:enumeration value="r"/>
            <xsd:enumeration value="dec"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextTabStop">
        <xsd:attribute name="pos" type="ST_Coordinate32" use="optional"/>
        <xsd:attribute name="algn" type="ST_TextTabAlignType" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TextTabStopList">
        <xsd:sequence>
            <xsd:element name="tab" type="CT_TextTabStop" minOccurs="0" maxOccurs="32"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TextLineBreak">
        <xsd:sequence>
            <xsd:element name="rPr" type="CT_TextCharacterProperties" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TextSpacing">
        <xsd:choice>
            <xsd:element name="spcPct" type="CT_TextSpacingPercent"/>
            <xsd:element name="spcPts" type="CT_TextSpacingPoint"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:simpleType name="ST_TextAlignType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="l"/>
            <xsd:enumeration value="ctr"/>
            <xsd:enumeration value="r"/>
            <xsd:enumeration value="just"/>
            <xsd:enumeration value="justLow"/>
            <xsd:enumeration value="dist"/>
            <xsd:enumeration value="thaiDist"/>
```



```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextFontAlignType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="auto"/>
            <xsd:enumeration value="t"/>
            <xsd:enumeration value="ctr"/>
            <xsd:enumeration value="base"/>
            <xsd:enumeration value="b"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextIndentLevelType">
        <xsd:restriction base="xsd:int">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="8"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextParagraphProperties">
        <xsd:sequence>
            <xsd:element name="lnSpc" type="CT_TextSpacing" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="spcBef" type="CT_TextSpacing" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="spcAft" type="CT_TextSpacing" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_TextBulletColor" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_TextBulletSize" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_TextBulletTypeface" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_TextBullet" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tabLst" type="CT_TextTabStopList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="defRPr" type="CT_TextCharacterProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="marL" type="ST_TextMargin" use="optional"/>
        <xsd:attribute name="marR" type="ST_TextMargin" use="optional"/>
        <xsd:attribute name="lvl" type="ST_TextIndentLevelType" use="optional"/>
        <xsd:attribute name="indent" type="ST_TextIndent" use="optional"/>
        <xsd:attribute name="algn" type="ST_TextAlignType" use="optional"/>
        <xsd:attribute name="defTabSz" type="ST_Coordinate32" use="optional"/>
        <xsd:attribute name="rtl" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="eaLnBrk" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="fontAlign" type="ST_TextFontAlignType" use="optional"/>
        <xsd:attribute name="latinLnBrk" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="hangingPunct" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TextField">
        <xsd:sequence>
            <xsd:element name="rPr" type="CT_TextCharacterProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pPr" type="CT_TextParagraphProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="t" type="xsd:string" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="s:ST_Guid" use="required"/>
        <xsd:attribute name="type" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:group name="EG_TextRun">
        <xsd:choice>
```

```
            <xsd:element name="r" type="CT_RegularTextRun"/>
            <xsd:element name="br" type="CT_TextLineBreak"/>
            <xsd:element name="fld" type="CT_TextField"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_RegularTextRun">
        <xsd:sequence>
            <xsd:element name="rPr" type="CT_TextCharacterProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="t" type="xsd:string" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

A.4.2 DrawingML - Picture

This schema is available in the file dml-picture.xsd.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.openxmlformats.org/drawingml/2006/picture"
  xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main" elementFormDefault="qualified"
  targetNamespace="http://schemas.openxmlformats.org/drawingml/2006/picture">
    <xsd:import namespace="http://schemas.openxmlformats.org/drawingml/2006/main" schemaLocation="dml-
      main.xsd"/>
    <xsd:complexType name="CT_PictureNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvPicPr" type="a:CT_NonVisualPictureProperties" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Picture">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element name="nvPicPr" type="CT_PictureNonVisual" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="blipFill" type="a:CT_BlipFillProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="pic" type="CT_Picture"/>
</xsd:schema>
```

A.4.3 DrawingML - Locked Canvas

This schema is available in the file dml-lockedCanvas.xsd.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.openxmlformats.org/drawingml/2006/lockedCanvas"
  xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
  xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
  elementFormDefault="qualified"
  targetNamespace="http://schemas.openxmlformats.org/drawingml/2006/lockedCanvas">
    <xsd:import namespace="http://schemas.openxmlformats.org/drawingml/2006/main" schemaLocation="dml-
      main.xsd"/>
    <xsd:element name="lockedCanvas" type="a:CT_GvmlGroupShape"/>
```



```
</xsd:schema>
```

A.4.4 DrawingML - WordprocessingML Drawing

This schema is available in the file dml-wordprocessingDrawing.xsd.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main"
xmlns:dpct="http://schemas.openxmlformats.org/drawingml/2006/picture"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xmlns="http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing"
targetNamespace="http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing"
elementFormDefault="qualified">

    <xsd:import namespace="http://schemas.openxmlformats.org/drawingml/2006/main" schemaLocation="dml-
      main.xsd"/>
    <xsd:import schemaLocation="wml.xsd"
      namespace="http://schemas.openxmlformats.org/wordprocessingml/2006/main"/>
    <xsd:import
      namespace="http://schemas.openxmlformats.org/drawingml/2006/picture"
      schemaLocation="dml-picture.xsd"/>
    <xsd:import
    namespace="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
     schemaLocation="shared-relationshipReference.xsd"/>
        <xsd:complexType name="CT_EffectExtent">
        <xsd:attribute name="l" type="a:ST_Coordinate" use="required"/>
        <xsd:attribute name="t" type="a:ST_Coordinate" use="required"/>
        <xsd:attribute name="r" type="a:ST_Coordinate" use="required"/>
        <xsd:attribute name="b" type="a:ST_Coordinate" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_WrapDistance">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_Inline">
        <xsd:sequence>
            <xsd:element name="extent" type="a:CT_PositiveSize2D"/>
            <xsd:element name="effectExtent" type="CT_EffectExtent" minOccurs="0"/>
            <xsd:element name="docPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvGraphicFramePr" type="a:CT_NonVisualGraphicFrameProperties"
              minOccurs="0" maxOccurs="1"/>
            <xsd:element ref="a:graphic" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="distT" type="ST_WrapDistance" use="optional"/>
        <xsd:attribute name="distB" type="ST_WrapDistance" use="optional"/>
        <xsd:attribute name="distL" type="ST_WrapDistance" use="optional"/>
        <xsd:attribute name="distR" type="ST_WrapDistance" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_WrapText">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="bothSides"/>
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="right"/>
```

```
            <xsd:enumeration value="largest"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_WrapPath">
        <xsd:sequence>
            <xsd:element name="start" type="a:CT_Point2D" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lineTo" type="a:CT_Point2D" minOccurs="2" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="edited" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WrapNone"/>
    <xsd:complexType name="CT_WrapSquare">
        <xsd:sequence>
            <xsd:element name="effectExtent" type="CT_EffectExtent" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="wrapText" type="ST_WrapText" use="required"/>
        <xsd:attribute name="distT" type="ST_WrapDistance" use="optional"/>
        <xsd:attribute name="distB" type="ST_WrapDistance" use="optional"/>
        <xsd:attribute name="distL" type="ST_WrapDistance" use="optional"/>
        <xsd:attribute name="distR" type="ST_WrapDistance" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WrapTight">
        <xsd:sequence>
            <xsd:element name="wrapPolygon" type="CT_WrapPath" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="wrapText" type="ST_WrapText" use="required"/>
        <xsd:attribute name="distL" type="ST_WrapDistance" use="optional"/>
        <xsd:attribute name="distR" type="ST_WrapDistance" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WrapThrough">
        <xsd:sequence>
            <xsd:element name="wrapPolygon" type="CT_WrapPath" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="wrapText" type="ST_WrapText" use="required"/>
        <xsd:attribute name="distL" type="ST_WrapDistance" use="optional"/>
        <xsd:attribute name="distR" type="ST_WrapDistance" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WrapTopBottom">
        <xsd:sequence>
            <xsd:element name="effectExtent" type="CT_EffectExtent" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="distT" type="ST_WrapDistance" use="optional"/>
        <xsd:attribute name="distB" type="ST_WrapDistance" use="optional"/>
    </xsd:complexType>
    <xsd:group name="EG_WrapType">
        <xsd:sequence>
            <xsd:choice minOccurs="1" maxOccurs="1">
                <xsd:element name="wrapNone" type="CT_WrapNone" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="wrapSquare" type="CT_WrapSquare" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="wrapTight" type="CT_WrapTight" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="wrapThrough" type="CT_WrapThrough" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="wrapTopAndBottom" type="CT_WrapTopBottom" minOccurs="1"
                  maxOccurs="1"/>
```



```
            </xsd:choice>
        </xsd:sequence>
    </xsd:group>
    <xsd:simpleType name="ST_PositionOffset">
        <xsd:restriction base="xsd:int"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_AlignH">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="right"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="inside"/>
            <xsd:enumeration value="outside"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_RelFromH">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="margin"/>
            <xsd:enumeration value="page"/>
            <xsd:enumeration value="column"/>
            <xsd:enumeration value="character"/>
            <xsd:enumeration value="leftMargin"/>
            <xsd:enumeration value="rightMargin"/>
            <xsd:enumeration value="insideMargin"/>
            <xsd:enumeration value="outsideMargin"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PosH">
        <xsd:sequence>
            <xsd:choice minOccurs="1" maxOccurs="1">
                <xsd:element name="align" type="ST_AlignH" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="posOffset" type="ST_PositionOffset" minOccurs="1" maxOccurs="1"/>
            </xsd:choice>
        </xsd:sequence>
        <xsd:attribute name="relativeFrom" type="ST_RelFromH" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_AlignV">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="top"/>
            <xsd:enumeration value="bottom"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="inside"/>
            <xsd:enumeration value="outside"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_RelFromV">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="margin"/>
            <xsd:enumeration value="page"/>
            <xsd:enumeration value="paragraph"/>
            <xsd:enumeration value="line"/>
            <xsd:enumeration value="topMargin"/>
            <xsd:enumeration value="bottomMargin"/>
```

```
            <xsd:enumeration value="insideMargin"/>
            <xsd:enumeration value="outsideMargin"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PosV">
        <xsd:sequence>
            <xsd:choice minOccurs="1" maxOccurs="1">
                <xsd:element name="align" type="ST_AlignV" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="posOffset" type="ST_PositionOffset" minOccurs="1" maxOccurs="1"/>
            </xsd:choice>
        </xsd:sequence>
        <xsd:attribute name="relativeFrom" type="ST_RelFromV" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Anchor">
        <xsd:sequence>
            <xsd:element name="simplePos" type="a:CT_Point2D"/>
            <xsd:element name="positionH" type="CT_PosH"/>
            <xsd:element name="positionV" type="CT_PosV"/>
            <xsd:element name="extent" type="a:CT_PositiveSize2D"/>
            <xsd:element name="effectExtent" type="CT_EffectExtent" minOccurs="0"/>
            <xsd:group ref="EG_WrapType"/>
            <xsd:element name="docPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvGraphicFramePr" type="a:CT_NonVisualGraphicFrameProperties"
              minOccurs="0" maxOccurs="1"/>
            <xsd:element ref="a:graphic" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="distT" type="ST_WrapDistance" use="optional"/>
        <xsd:attribute name="distB" type="ST_WrapDistance" use="optional"/>
        <xsd:attribute name="distL" type="ST_WrapDistance" use="optional"/>
        <xsd:attribute name="distR" type="ST_WrapDistance" use="optional"/>
        <xsd:attribute name="simplePos" type="xsd:boolean"/>
        <xsd:attribute name="relativeHeight" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="behindDoc" type="xsd:boolean" use="required"/>
        <xsd:attribute name="locked" type="xsd:boolean" use="required"/>
        <xsd:attribute name="layoutInCell" type="xsd:boolean" use="required"/>
        <xsd:attribute name="hidden" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="allowOverlap" type="xsd:boolean" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TxbxContent">
        <xsd:group ref="w:EG_BlockLevelElts" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TextboxInfo">
        <xsd:sequence>
            <xsd:element name="txbxContent" type="CT_TxbxContent" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:unsignedShort" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_LinkedTextboxInformation">
        <xsd:sequence>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
```



```
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:unsignedShort" use="required"/>
        <xsd:attribute name="seq" type="xsd:unsignedShort" use="required"/>
    </xsd:complexType>
            <xsd:complexType name="CT_WordprocessingShape">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="0" maxOccurs="1"/>
            <xsd:choice minOccurs="1" maxOccurs="1">
              <xsd:element name="cNvSpPr" type="a:CT_NonVisualDrawingShapeProps" minOccurs="1"
              maxOccurs="1"/>
              <xsd:element name="cNvCnPr" type="a:CT_NonVisualConnectorProperties" minOccurs="1"
              maxOccurs="1"/>
            </xsd:choice>
              <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
              <xsd:element name="style" type="a:CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
              <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
              <xsd:choice minOccurs="0" maxOccurs="1">
              <xsd:element name="txbx" type="CT_TextboxInfo" minOccurs="1" maxOccurs="1"/>
              <xsd:element name="linkedTxbx" type="CT_LinkedTextboxInformation" minOccurs="1"
              maxOccurs="1"/>
            </xsd:choice>
            <xsd:element name="bodyPr" type="a:CT_TextBodyProperties" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="normalEastAsianFlow" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GraphicFrame">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvFrPr" type="a:CT_NonVisualGraphicFrameProperties" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="xfrm" type="a:CT_Transform2D" minOccurs="1" maxOccurs="1"/>
            <xsd:element ref="a:graphic" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_WordprocessingContentPartNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cNvContentPartPr" type="a:CT_NonVisualContentPartProperties"
              minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_WordprocessingContentPart">
        <xsd:sequence>
            <xsd:element name="nvContentPartPr" type="CT_WordprocessingContentPartNonVisual"
              minOccurs="0" maxOccurs="1"/>
            <xsd:element name="xfrm" type="a:CT_Transform2D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="bwMode" type="a:ST_BlackWhiteMode" use="optional"/>
```

```
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WordprocessingGroup">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cNvGrpSpPr" type="a:CT_NonVisualGroupDrawingShapeProps" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="grpSpPr" type="a:CT_GroupShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
              <xsd:element ref="wsp"/>
              <xsd:element name="grpSp" type="CT_WordprocessingGroup"/>
              <xsd:element name="graphicFrame" type="CT_GraphicFrame"/>
              <xsd:element ref="dpct:pic"/>
              <xsd:element name="contentPart" type="CT_WordprocessingContentPart"/>
            </xsd:choice>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_WordprocessingCanvas">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element name="bg" type="a:CT_BackgroundFormatting" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="whole" type="a:CT_WholeE2oFormatting" minOccurs="0" maxOccurs="1"/>
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
              <xsd:element ref="wsp"/>
              <xsd:element ref="dpct:pic"/>
              <xsd:element name="contentPart" type="CT_WordprocessingContentPart"/>
              <xsd:element ref="wgp"/>
              <xsd:element name="graphicFrame" type="CT_GraphicFrame"/>
            </xsd:choice>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="wpc" type="CT_WordprocessingCanvas"/>
    <xsd:element name="wgp" type="CT_WordprocessingGroup"/>
    <xsd:element name="wsp" type="CT_WordprocessingShape"/>
    <xsd:element name="inline" type="CT_Inline"/>
    <xsd:element name="anchor" type="CT_Anchor"/>
</xsd:schema>
```

A.4.5 DrawingML - SpreadsheetML Drawing

This schema is available in the file dml-spreadsheetDrawing.xsd.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
  xmlns="http://schemas.openxmlformats.org/drawingml/2006/spreadsheetDrawing"
  xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
  targetNamespace="http://schemas.openxmlformats.org/drawingml/2006/spreadsheetDrawing"
  elementFormDefault="qualified">
    <xsd:import namespace="http://schemas.openxmlformats.org/drawingml/2006/main" schemaLocation="dml-
      main.xsd"/>
```



```
    <xsd:import schemaLocation="shared-relationshipReference.xsd"
      namespace="http://schemas.openxmlformats.org/officeDocument/2006/relationships"/>
    <xsd:element name="from" type="CT_Marker"/>
    <xsd:element name="to" type="CT_Marker"/>
    <xsd:complexType name="CT_AnchorClientData">
        <xsd:attribute name="fLocksWithSheet" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="fPrintsWithSheet" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ShapeNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvSpPr" type="a:CT_NonVisualDrawingShapeProps" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Shape">
        <xsd:sequence>
            <xsd:element name="nvSpPr" type="CT_ShapeNonVisual" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="style" type="a:CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="txBody" type="a:CT_TextBody" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="macro" type="xsd:string" use="optional"/>
        <xsd:attribute name="textlink" type="xsd:string" use="optional"/>
        <xsd:attribute name="fLocksText" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="fPublished" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ConnectorNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvCxnSpPr" type="a:CT_NonVisualConnectorProperties" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Connector">
        <xsd:sequence>
            <xsd:element name="nvCxnSpPr" type="CT_ConnectorNonVisual" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="style" type="a:CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="macro" type="xsd:string" use="optional"/>
        <xsd:attribute name="fPublished" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PictureNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvPicPr" type="a:CT_NonVisualPictureProperties" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Picture">
        <xsd:sequence>
            <xsd:element name="nvPicPr" type="CT_PictureNonVisual" minOccurs="1" maxOccurs="1"/>
```

```
            <xsd:element name="blipFill" type="a:CT_BlipFillProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="style" type="a:CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="macro" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="fPublished" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GraphicalObjectFrameNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvGraphicFramePr" type="a:CT_NonVisualGraphicFrameProperties"
              minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GraphicalObjectFrame">
        <xsd:sequence>
            <xsd:element name="nvGraphicFramePr" type="CT_GraphicalObjectFrameNonVisual" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="xfrm" type="a:CT_Transform2D" minOccurs="1" maxOccurs="1"/>
            <xsd:element ref="a:graphic" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="macro" type="xsd:string" use="optional"/>
        <xsd:attribute name="fPublished" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupShapeNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvGrpSpPr" type="a:CT_NonVisualGroupDrawingShapeProps" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupShape">
        <xsd:sequence>
            <xsd:element name="nvGrpSpPr" type="CT_GroupShapeNonVisual" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="grpSpPr" type="a:CT_GroupShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="sp" type="CT_Shape"/>
                <xsd:element name="grpSp" type="CT_GroupShape"/>
                <xsd:element name="graphicFrame" type="CT_GraphicalObjectFrame"/>
                <xsd:element name="cxnSp" type="CT_Connector"/>
                <xsd:element name="pic" type="CT_Picture"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_ObjectChoices">
        <xsd:sequence>
            <xsd:choice minOccurs="1" maxOccurs="1">
                <xsd:element name="sp" type="CT_Shape"/>
                <xsd:element name="grpSp" type="CT_GroupShape"/>
                <xsd:element name="graphicFrame" type="CT_GraphicalObjectFrame"/>
                <xsd:element name="cxnSp" type="CT_Connector"/>
                <xsd:element name="pic" type="CT_Picture"/>
                <xsd:element name="contentPart" type="CT_Rel"/>
```



```
            </xsd:choice>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_Rel">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_ColID">
        <xsd:restriction base="xsd:int">
            <xsd:minInclusive value="0"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_RowID">
        <xsd:restriction base="xsd:int">
            <xsd:minInclusive value="0"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Marker">
        <xsd:sequence>
            <xsd:element name="col" type="ST_ColID"/>
            <xsd:element name="colOff" type="a:ST_Coordinate"/>
            <xsd:element name="row" type="ST_RowID"/>
            <xsd:element name="rowOff" type="a:ST_Coordinate"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_EditAs">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="twoCell"/>
            <xsd:enumeration value="oneCell"/>
            <xsd:enumeration value="absolute"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TwoCellAnchor">
        <xsd:sequence>
            <xsd:element name="from" type="CT_Marker"/>
            <xsd:element name="to" type="CT_Marker"/>
            <xsd:group ref="EG_ObjectChoices"/>
            <xsd:element name="clientData" type="CT_AnchorClientData" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="editAs" type="ST_EditAs" use="optional" default="twoCell"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OneCellAnchor">
        <xsd:sequence>
            <xsd:element name="from" type="CT_Marker"/>
            <xsd:element name="ext" type="a:CT_PositiveSize2D"/>
            <xsd:group ref="EG_ObjectChoices"/>
            <xsd:element name="clientData" type="CT_AnchorClientData" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_AbsoluteAnchor">
        <xsd:sequence>
            <xsd:element name="pos" type="a:CT_Point2D"/>
            <xsd:element name="ext" type="a:CT_PositiveSize2D"/>
            <xsd:group ref="EG_ObjectChoices"/>
```

```
            <xsd:element name="clientData" type="CT_AnchorClientData" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_Anchor">
        <xsd:choice>
            <xsd:element name="twoCellAnchor" type="CT_TwoCellAnchor"/>
            <xsd:element name="oneCellAnchor" type="CT_OneCellAnchor"/>
            <xsd:element name="absoluteAnchor" type="CT_AbsoluteAnchor"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_Drawing">
        <xsd:sequence>
            <xsd:group ref="EG_Anchor" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="wsDr" type="CT_Drawing"/>
</xsd:schema>
```

A.5 DrawingML - Components

A.5.1 DrawingML - Charts

This schema is available in the file dml-chart.xsd.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
  xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
  xmlns="http://schemas.openxmlformats.org/drawingml/2006/chart"
  xmlns:cdr="http://schemas.openxmlformats.org/drawingml/2006/chartDrawing"
  xmlns:s="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
  targetNamespace="http://schemas.openxmlformats.org/drawingml/2006/chart"
  elementFormDefault="qualified" attributeFormDefault="unqualified" blockDefault="#all">
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
      schemaLocation="shared-relationshipReference.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/drawingml/2006/main" schemaLocation="dml-
      main.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/drawingml/2006/chartDrawing"
      schemaLocation="dml-chartDrawing.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
    <xsd:complexType name="CT_Boolean">
        <xsd:attribute name="val" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Double">
        <xsd:attribute name="val" type="xsd:double" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_UnsignedInt">
        <xsd:attribute name="val" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RelId">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Extension">
```



```
        <xsd:sequence>
            <xsd:any processContents="lax"/>
        </xsd:sequence>
        <xsd:attribute name="uri" type="xsd:token"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ExtensionList">
        <xsd:sequence>
            <xsd:element name="ext" type="CT_Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_NumVal">
        <xsd:sequence>
            <xsd:element name="v" type="s:ST_Xstring" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="idx" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="formatCode" type="s:ST_Xstring" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NumData">
        <xsd:sequence>
            <xsd:element name="formatCode" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ptCount" type="CT_UnsignedInt" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pt" type="CT_NumVal" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_NumRef">
        <xsd:sequence>
            <xsd:element name="f" type="xsd:string" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="numCache" type="CT_NumData" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_NumDataSource">
        <xsd:sequence>
            <xsd:choice minOccurs="1" maxOccurs="1">
                <xsd:element name="numRef" type="CT_NumRef" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="numLit" type="CT_NumData" minOccurs="1" maxOccurs="1"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_StrVal">
        <xsd:sequence>
            <xsd:element name="v" type="s:ST_Xstring" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="idx" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_StrData">
        <xsd:sequence>
            <xsd:element name="ptCount" type="CT_UnsignedInt" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pt" type="CT_StrVal" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
```

```
    <xsd:complexType name="CT_StrRef">
        <xsd:sequence>
            <xsd:element name="f" type="xsd:string" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="strCache" type="CT_StrData" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Tx">
        <xsd:sequence>
            <xsd:choice minOccurs="1" maxOccurs="1">
                <xsd:element name="strRef" type="CT_StrRef" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="rich" type="a:CT_TextBody" minOccurs="1" maxOccurs="1"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TextLanguageID">
        <xsd:attribute name="val" type="s:ST_Lang" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Lvl">
        <xsd:sequence>
            <xsd:element name="pt" type="CT_StrVal" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_MultiLvlStrData">
        <xsd:sequence>
            <xsd:element name="ptCount" type="CT_UnsignedInt" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="lvl" type="CT_Lvl" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_MultiLvlStrRef">
        <xsd:sequence>
            <xsd:element name="f" type="xsd:string" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="multiLvlStrCache" type="CT_MultiLvlStrData" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_AxDataSource">
        <xsd:sequence>
            <xsd:choice minOccurs="1" maxOccurs="1">
                <xsd:element name="multiLvlStrRef" type="CT_MultiLvlStrRef" minOccurs="1"
                  maxOccurs="1"/>
                <xsd:element name="numRef" type="CT_NumRef" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="numLit" type="CT_NumData" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="strRef" type="CT_StrRef" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="strLit" type="CT_StrData" minOccurs="1" maxOccurs="1"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SerTx">
        <xsd:sequence>
            <xsd:choice minOccurs="1" maxOccurs="1">
```



```
                <xsd:element name="strRef" type="CT_StrRef" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="v" type="s:ST_Xstring" minOccurs="1" maxOccurs="1"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_LayoutTarget">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="inner"/>
            <xsd:enumeration value="outer"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_LayoutTarget">
        <xsd:attribute name="val" type="ST_LayoutTarget" default="outer"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_LayoutMode">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="edge"/>
            <xsd:enumeration value="factor"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_LayoutMode">
        <xsd:attribute name="val" type="ST_LayoutMode" default="factor"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ManualLayout">
        <xsd:sequence>
            <xsd:element name="layoutTarget" type="CT_LayoutTarget" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="xMode" type="CT_LayoutMode" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="yMode" type="CT_LayoutMode" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="wMode" type="CT_LayoutMode" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="hMode" type="CT_LayoutMode" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="x" type="CT_Double" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="y" type="CT_Double" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="w" type="CT_Double" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="h" type="CT_Double" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Layout">
        <xsd:sequence>
            <xsd:element name="manualLayout" type="CT_ManualLayout" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Title">
        <xsd:sequence>
            <xsd:element name="tx" type="CT_Tx" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="layout" type="CT_Layout" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="overlay" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="txPr" type="a:CT_TextBody" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
```

```
    <xsd:simpleType name="ST_RotX">
        <xsd:restriction base="xsd:byte">
            <xsd:minInclusive value="-90"/>
            <xsd:maxInclusive value="90"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_RotX">
        <xsd:attribute name="val" type="ST_RotX" default="0"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_HPercent">
        <xsd:restriction base="xsd:unsignedShort">
            <xsd:minInclusive value="5"/>
            <xsd:maxInclusive value="500"/>
        </xsd:restriction>
        <xsd:union memberTypes="ST_HPercentWithSymbol ST_HPercentUShort"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_HPercentWithSymbol">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="0*(([5-9])|([1-9][0-9])|([1-4][0-9][0-9])|500)%"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_HPercentUShort">
        <xsd:restriction base="xsd:unsignedShort">
            <xsd:minInclusive value="5"/>
            <xsd:maxInclusive value="500"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_HPercent">
        <xsd:attribute name="val" type="ST_HPercent" default="100%"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_RotY">
        <xsd:restriction base="xsd:unsignedShort">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="360"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_RotY">
        <xsd:attribute name="val" type="ST_RotY" default="0"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DepthPercent">
        <xsd:union memberTypes="ST_DepthPercentWithSymbol ST_DepthPercentUShort"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_DepthPercentWithSymbol">
        <xsd:restriction base="xsd:string">
          <xsd:pattern value="0*(([2-9][0-9])|([1-9][0-9][0-9])|(1[0-9][0-9][0-9])|2000)%"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_DepthPercentUShort">
        <xsd:restriction base="xsd:unsignedShort">
          <xsd:minInclusive value="20"/>
          <xsd:maxInclusive value="2000"/>
        </xsd:restriction>
    </xsd:simpleType>
```



```
    <xsd:complexType name="CT_DepthPercent">
        <xsd:attribute name="val" type="ST_DepthPercent" default="100%"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Perspective">
        <xsd:restriction base="xsd:unsignedByte">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="240"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Perspective">
        <xsd:attribute name="val" type="ST_Perspective" default="30"/>
    </xsd:complexType>
    <xsd:complexType name="CT_View3D">
        <xsd:sequence>
            <xsd:element name="rotX" type="CT_RotX" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="hPercent" type="CT_HPercent" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="rotY" type="CT_RotY" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="depthPercent" type="CT_DepthPercent" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="rAngAx" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="perspective" type="CT_Perspective" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Surface">
        <xsd:sequence>
            <xsd:element name="thickness" type="CT_Thickness" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pictureOptions" type="CT_PictureOptions" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_Thickness">
        <xsd:union memberTypes="ST_ThicknessPercent xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ThicknessPercent">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="([0-9]+)%"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Thickness">
        <xsd:attribute name="val" type="ST_Thickness" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DTable">
        <xsd:sequence>
            <xsd:element name="showHorzBorder" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="showVertBorder" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="showOutline" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="showKeys" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="txPr" type="a:CT_TextBody" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
```

```
    <xsd:simpleType name="ST_GapAmount">
        <xsd:union memberTypes="ST_GapAmountPercent ST_GapAmountUShort"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_GapAmountPercent">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="0*(([0-9])|([1-9][0-9])|([1-4][0-9][0-9])|500)%"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_GapAmountUShort">
        <xsd:restriction base="xsd:unsignedShort">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="500"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_GapAmount">
        <xsd:attribute name="val" type="ST_GapAmount" default="150%"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Overlap">
        <xsd:union memberTypes="ST_OverlapPercent ST_OverlapByte"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_OverlapPercent">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="(-?0*(([0-9])|([1-9][0-9])|100))%"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_OverlapByte">
        <xsd:restriction base="xsd:byte">
            <xsd:minInclusive value="-100"/>
            <xsd:maxInclusive value="100"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Overlap">
        <xsd:attribute name="val" type="ST_Overlap" default="0%"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_BubbleScale">
        <xsd:union memberTypes="ST_BubbleScalePercent ST_BubbleScaleUInt"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_BubbleScalePercent">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="0*(([0-9])|([1-9][0-9])|([1-2][0-9][0-9])|300)%"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_BubbleScaleUInt">
        <xsd:restriction base="xsd:unsignedInt">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="300"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_BubbleScale">
        <xsd:attribute name="val" type="ST_BubbleScale" default="100%"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SizeRepresents">
        <xsd:restriction base="xsd:string">
```



```
            <xsd:enumeration value="area"/>
            <xsd:enumeration value="w"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SizeRepresents">
        <xsd:attribute name="val" type="ST_SizeRepresents" default="area"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_FirstSliceAng">
        <xsd:restriction base="xsd:unsignedShort">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="360"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_FirstSliceAng">
        <xsd:attribute name="val" type="ST_FirstSliceAng" default="0"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_HoleSize">
        <xsd:union memberTypes="ST_HoleSizePercent ST_HoleSizeUByte"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_HoleSizePercent">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="0*([1-9]|([1-8][0-9])|90)%"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_HoleSizeUByte">
        <xsd:restriction base="xsd:unsignedByte">
            <xsd:minInclusive value="10"/>
            <xsd:maxInclusive value="90"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_HoleSize">
        <xsd:attribute name="val" type="ST_HoleSize" default="10%"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SplitType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="auto"/>
            <xsd:enumeration value="cust"/>
            <xsd:enumeration value="percent"/>
            <xsd:enumeration value="pos"/>
            <xsd:enumeration value="val"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SplitType">
        <xsd:attribute name="val" type="ST_SplitType" default="auto"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustSplit">
        <xsd:sequence>
            <xsd:element name="secondPiePt" type="CT_UnsignedInt" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_SecondPieSize">
        <xsd:union memberTypes="ST_SecondPieSizePercent ST_SecondPieSizeUShort"/>
```

```
    </xsd:simpleType>
    <xsd:simpleType name="ST_SecondPieSizePercent">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="0*(([5-9])|([1-9][0-9])|(1[0-9][0-9])|200)%"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_SecondPieSizeUShort">
        <xsd:restriction base="xsd:unsignedShort">
            <xsd:minInclusive value="5"/>
            <xsd:maxInclusive value="200"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SecondPieSize">
        <xsd:attribute name="val" type="ST_SecondPieSize" default="75%"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NumFmt">
        <xsd:attribute name="formatCode" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="sourceLinked" type="xsd:boolean"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_LblAlgn">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="ctr"/>
            <xsd:enumeration value="l"/>
            <xsd:enumeration value="r"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_LblAlgn">
        <xsd:attribute name="val" type="ST_LblAlgn" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DLblPos">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="bestFit"/>
            <xsd:enumeration value="b"/>
            <xsd:enumeration value="ctr"/>
            <xsd:enumeration value="inBase"/>
            <xsd:enumeration value="inEnd"/>
            <xsd:enumeration value="l"/>
            <xsd:enumeration value="outEnd"/>
            <xsd:enumeration value="r"/>
            <xsd:enumeration value="t"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DLblPos">
        <xsd:attribute name="val" type="ST_DLblPos" use="required"/>
    </xsd:complexType>
    <xsd:group name="EG_DLblShared">
        <xsd:sequence>
            <xsd:element name="numFmt" type="CT_NumFmt" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="txPr" type="a:CT_TextBody" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dLblPos" type="CT_DLblPos" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="showLegendKey" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="showVal" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
```



```
            <xsd:element name="showCatName" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="showSerName" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="showPercent" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="showBubbleSize" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="separator" type="xsd:string" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:group name="Group_DLbl">
        <xsd:sequence>
            <xsd:element name="layout" type="CT_Layout" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tx" type="CT_Tx" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_DLblShared" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_DLbl">
        <xsd:sequence>
            <xsd:element name="idx" type="CT_UnsignedInt" minOccurs="1" maxOccurs="1"/>
            <xsd:choice>
                <xsd:element name="delete" type="CT_Boolean" minOccurs="1" maxOccurs="1"/>
                <xsd:group ref="Group_DLbl" minOccurs="1" maxOccurs="1"/>
            </xsd:choice>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="Group_DLbls">
        <xsd:sequence>
            <xsd:group ref="EG_DLblShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="showLeaderLines" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="leaderLines" type="CT_ChartLines" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_DLbls">
        <xsd:sequence>
            <xsd:element name="dLbl" type="CT_DLbl" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:choice>
                <xsd:element name="delete" type="CT_Boolean" minOccurs="1" maxOccurs="1"/>
                <xsd:group ref="Group_DLbls" minOccurs="1" maxOccurs="1"/>
            </xsd:choice>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_MarkerStyle">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="circle"/>
            <xsd:enumeration value="dash"/>
            <xsd:enumeration value="diamond"/>
            <xsd:enumeration value="dot"/>
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="picture"/>
            <xsd:enumeration value="plus"/>
            <xsd:enumeration value="square"/>
            <xsd:enumeration value="star"/>
            <xsd:enumeration value="triangle"/>
```

```
            <xsd:enumeration value="x"/>
            <xsd:enumeration value="auto"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_MarkerStyle">
        <xsd:attribute name="val" type="ST_MarkerStyle" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_MarkerSize">
        <xsd:restriction base="xsd:unsignedByte">
            <xsd:minInclusive value="2"/>
            <xsd:maxInclusive value="72"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_MarkerSize">
        <xsd:attribute name="val" type="ST_MarkerSize" default="5"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Marker">
        <xsd:sequence>
            <xsd:element name="symbol" type="CT_MarkerStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="size" type="CT_MarkerSize" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DPt">
        <xsd:sequence>
            <xsd:element name="idx" type="CT_UnsignedInt" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="invertIfNegative" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="marker" type="CT_Marker" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bubble3D" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="explosion" type="CT_UnsignedInt" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pictureOptions" type="CT_PictureOptions" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_TrendlineType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="exp"/>
            <xsd:enumeration value="linear"/>
            <xsd:enumeration value="log"/>
            <xsd:enumeration value="movingAvg"/>
            <xsd:enumeration value="poly"/>
            <xsd:enumeration value="power"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TrendlineType">
        <xsd:attribute name="val" type="ST_TrendlineType" default="linear"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Order">
        <xsd:restriction base="xsd:unsignedByte">
            <xsd:minInclusive value="2"/>
            <xsd:maxInclusive value="6"/>
```



```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Order">
        <xsd:attribute name="val" type="ST_Order" default="2"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Period">
        <xsd:restriction base="xsd:unsignedInt">
            <xsd:minInclusive value="2"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Period">
        <xsd:attribute name="val" type="ST_Period" default="2"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TrendlineLbl">
        <xsd:sequence>
            <xsd:element name="layout" type="CT_Layout" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tx" type="CT_Tx" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="numFmt" type="CT_NumFmt" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="txPr" type="a:CT_TextBody" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Trendline">
        <xsd:sequence>
            <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="trendlineType" type="CT_TrendlineType" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="order" type="CT_Order" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="period" type="CT_Period" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="forward" type="CT_Double" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="backward" type="CT_Double" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="intercept" type="CT_Double" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dispRSqr" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dispEq" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="trendlineLbl" type="CT_TrendlineLbl" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_ErrDir">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="x"/>
            <xsd:enumeration value="y"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_ErrDir">
        <xsd:attribute name="val" type="ST_ErrDir" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_ErrBarType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="both"/>
            <xsd:enumeration value="minus"/>
            <xsd:enumeration value="plus"/>
```

```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_ErrBarType">
        <xsd:attribute name="val" type="ST_ErrBarType" default="both"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_ErrValType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="cust"/>
            <xsd:enumeration value="fixedVal"/>
            <xsd:enumeration value="percentage"/>
            <xsd:enumeration value="stdDev"/>
            <xsd:enumeration value="stdErr"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_ErrValType">
        <xsd:attribute name="val" type="ST_ErrValType" default="fixedVal"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ErrBars">
        <xsd:sequence>
            <xsd:element name="errDir" type="CT_ErrDir" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="errBarType" type="CT_ErrBarType" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="errValType" type="CT_ErrValType" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="noEndCap" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="plus" type="CT_NumDataSource" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="minus" type="CT_NumDataSource" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="val" type="CT_Double" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_UpDownBar">
        <xsd:sequence>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_UpDownBars">
        <xsd:sequence>
            <xsd:element name="gapWidth" type="CT_GapAmount" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="upBars" type="CT_UpDownBar" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="downBars" type="CT_UpDownBar" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_SerShared">
        <xsd:sequence>
            <xsd:element name="idx" type="CT_UnsignedInt" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="order" type="CT_UnsignedInt" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="tx" type="CT_SerTx" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_LineSer">
        <xsd:sequence>
```



```
            <xsd:group ref="EG_SerShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="marker" type="CT_Marker" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dPt" type="CT_DPt" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="dLbls" type="CT_DLbls" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="trendline" type="CT_Trendline" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="errBars" type="CT_ErrBars" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cat" type="CT_AxDataSource" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="val" type="CT_NumDataSource" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="smooth" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ScatterSer">
        <xsd:sequence>
            <xsd:group ref="EG_SerShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="marker" type="CT_Marker" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dPt" type="CT_DPt" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="dLbls" type="CT_DLbls" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="trendline" type="CT_Trendline" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="errBars" type="CT_ErrBars" minOccurs="0" maxOccurs="2"/>
            <xsd:element name="xVal" type="CT_AxDataSource" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="yVal" type="CT_NumDataSource" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="smooth" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_RadarSer">
        <xsd:sequence>
            <xsd:group ref="EG_SerShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="marker" type="CT_Marker" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dPt" type="CT_DPt" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="dLbls" type="CT_DLbls" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cat" type="CT_AxDataSource" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="val" type="CT_NumDataSource" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_BarSer">
        <xsd:sequence>
            <xsd:group ref="EG_SerShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="invertIfNegative" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pictureOptions" type="CT_PictureOptions" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dPt" type="CT_DPt" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="dLbls" type="CT_DLbls" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="trendline" type="CT_Trendline" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="errBars" type="CT_ErrBars" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cat" type="CT_AxDataSource" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="val" type="CT_NumDataSource" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="shape" type="CT_Shape" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_AreaSer">
```

```
        <xsd:sequence>
            <xsd:group ref="EG_SerShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="pictureOptions" type="CT_PictureOptions" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dPt" type="CT_DPt" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="dLbls" type="CT_DLbls" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="trendline" type="CT_Trendline" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="errBars" type="CT_ErrBars" minOccurs="0" maxOccurs="2"/>
            <xsd:element name="cat" type="CT_AxDataSource" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="val" type="CT_NumDataSource" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_PieSer">
        <xsd:sequence>
            <xsd:group ref="EG_SerShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="explosion" type="CT_UnsignedInt" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dPt" type="CT_DPt" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="dLbls" type="CT_DLbls" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cat" type="CT_AxDataSource" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="val" type="CT_NumDataSource" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_BubbleSer">
        <xsd:sequence>
            <xsd:group ref="EG_SerShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="invertIfNegative" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dPt" type="CT_DPt" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="dLbls" type="CT_DLbls" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="trendline" type="CT_Trendline" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="errBars" type="CT_ErrBars" minOccurs="0" maxOccurs="2"/>
            <xsd:element name="xVal" type="CT_AxDataSource" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="yVal" type="CT_NumDataSource" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bubbleSize" type="CT_NumDataSource" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bubble3D" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SurfaceSer">
        <xsd:sequence>
            <xsd:group ref="EG_SerShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cat" type="CT_AxDataSource" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="val" type="CT_NumDataSource" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_Grouping">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="percentStacked"/>
            <xsd:enumeration value="standard"/>
            <xsd:enumeration value="stacked"/>
        </xsd:restriction>
    </xsd:simpleType>
```



```
    <xsd:complexType name="CT_Grouping">
        <xsd:attribute name="val" type="ST_Grouping" default="standard"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ChartLines">
        <xsd:sequence>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_LineChartShared">
        <xsd:sequence>
            <xsd:element name="grouping" type="CT_Grouping" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="varyColors" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ser" type="CT_LineSer" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="dLbls" type="CT_DLbls" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dropLines" type="CT_ChartLines" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_LineChart">
        <xsd:sequence>
            <xsd:group ref="EG_LineChartShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="hiLowLines" type="CT_ChartLines" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="upDownBars" type="CT_UpDownBars" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="marker" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="smooth" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="axId" type="CT_UnsignedInt" minOccurs="2" maxOccurs="2"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Line3DChart">
        <xsd:sequence>
            <xsd:group ref="EG_LineChartShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="gapDepth" type="CT_GapAmount" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="axId" type="CT_UnsignedInt" minOccurs="3" maxOccurs="3"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_StockChart">
        <xsd:sequence>
            <xsd:element name="ser" type="CT_LineSer" minOccurs="3" maxOccurs="4"/>
            <xsd:element name="dLbls" type="CT_DLbls" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dropLines" type="CT_ChartLines" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="hiLowLines" type="CT_ChartLines" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="upDownBars" type="CT_UpDownBars" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="axId" type="CT_UnsignedInt" minOccurs="2" maxOccurs="2"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_ScatterStyle">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="line"/>
            <xsd:enumeration value="lineMarker"/>
            <xsd:enumeration value="marker"/>
```



```
            <xsd:enumeration value="smooth"/>
            <xsd:enumeration value="smoothMarker"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_ScatterStyle">
        <xsd:attribute name="val" type="ST_ScatterStyle" default="marker"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ScatterChart">
        <xsd:sequence>
            <xsd:element name="scatterStyle" type="CT_ScatterStyle" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="varyColors" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ser" type="CT_ScatterSer" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="dLbls" type="CT_DLbls" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="axId" type="CT_UnsignedInt" minOccurs="2" maxOccurs="2"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_RadarStyle">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="standard"/>
            <xsd:enumeration value="marker"/>
            <xsd:enumeration value="filled"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_RadarStyle">
        <xsd:attribute name="val" type="ST_RadarStyle" default="standard"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RadarChart">
        <xsd:sequence>
            <xsd:element name="radarStyle" type="CT_RadarStyle" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="varyColors" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ser" type="CT_RadarSer" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="dLbls" type="CT_DLbls" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="axId" type="CT_UnsignedInt" minOccurs="2" maxOccurs="2"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_BarGrouping">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="percentStacked"/>
            <xsd:enumeration value="clustered"/>
            <xsd:enumeration value="standard"/>
            <xsd:enumeration value="stacked"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_BarGrouping">
        <xsd:attribute name="val" type="ST_BarGrouping" default="clustered"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_BarDir">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="bar"/>
            <xsd:enumeration value="col"/>
        </xsd:restriction>
```



```
    </xsd:simpleType>
    <xsd:complexType name="CT_BarDir">
        <xsd:attribute name="val" type="ST_BarDir" default="col"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Shape">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="cone"/>
            <xsd:enumeration value="coneToMax"/>
            <xsd:enumeration value="box"/>
            <xsd:enumeration value="cylinder"/>
            <xsd:enumeration value="pyramid"/>
            <xsd:enumeration value="pyramidToMax"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Shape">
        <xsd:attribute name="val" type="ST_Shape" default="box"/>
    </xsd:complexType>
    <xsd:group name="EG_BarChartShared">
        <xsd:sequence>
            <xsd:element name="barDir" type="CT_BarDir" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="grouping" type="CT_BarGrouping" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="varyColors" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ser" type="CT_BarSer" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="dLbls" type="CT_DLbls" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_BarChart">
        <xsd:sequence>
            <xsd:group ref="EG_BarChartShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="gapWidth" type="CT_GapAmount" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="overlap" type="CT_Overlap" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="serLines" type="CT_ChartLines" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="axId" type="CT_UnsignedInt" minOccurs="2" maxOccurs="2"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Bar3DChart">
        <xsd:sequence>
            <xsd:group ref="EG_BarChartShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="gapWidth" type="CT_GapAmount" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="gapDepth" type="CT_GapAmount" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="shape" type="CT_Shape" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="axId" type="CT_UnsignedInt" minOccurs="2" maxOccurs="3"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_AreaChartShared">
        <xsd:sequence>
            <xsd:element name="grouping" type="CT_Grouping" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="varyColors" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ser" type="CT_AreaSer" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="dLbls" type="CT_DLbls" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dropLines" type="CT_ChartLines" minOccurs="0" maxOccurs="1"/>
```

```
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_AreaChart">
        <xsd:sequence>
            <xsd:group ref="EG_AreaChartShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="axId" type="CT_UnsignedInt" minOccurs="2" maxOccurs="2"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Area3DChart">
        <xsd:sequence>
            <xsd:group ref="EG_AreaChartShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="gapDepth" type="CT_GapAmount" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="axId" type="CT_UnsignedInt" minOccurs="2" maxOccurs="3"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_PieChartShared">
        <xsd:sequence>
            <xsd:element name="varyColors" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ser" type="CT_PieSer" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="dLbls" type="CT_DLbls" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_PieChart">
        <xsd:sequence>
            <xsd:group ref="EG_PieChartShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="firstSliceAng" type="CT_FirstSliceAng" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Pie3DChart">
        <xsd:sequence>
            <xsd:group ref="EG_PieChartShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DoughnutChart">
        <xsd:sequence>
            <xsd:group ref="EG_PieChartShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="firstSliceAng" type="CT_FirstSliceAng" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="holeSize" type="CT_HoleSize" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_OfPieType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="pie"/>
            <xsd:enumeration value="bar"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_OfPieType">
        <xsd:attribute name="val" type="ST_OfPieType" default="pie"/>
```



```
    </xsd:complexType>
    <xsd:complexType name="CT_OfPieChart">
        <xsd:sequence>
            <xsd:element name="ofPieType" type="CT_OfPieType" minOccurs="1" maxOccurs="1"/>
            <xsd:group ref="EG_PieChartShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="gapWidth" type="CT_GapAmount" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="splitType" type="CT_SplitType" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="splitPos" type="CT_Double" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="custSplit" type="CT_CustSplit" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="secondPieSize" type="CT_SecondPieSize" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="serLines" type="CT_ChartLines" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_BubbleChart">
        <xsd:sequence>
            <xsd:element name="varyColors" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ser" type="CT_BubbleSer" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="dLbls" type="CT_DLbls" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bubble3D" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bubbleScale" type="CT_BubbleScale" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="showNegBubbles" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sizeRepresents" type="CT_SizeRepresents" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="axId" type="CT_UnsignedInt" minOccurs="2" maxOccurs="2"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_BandFmt">
        <xsd:sequence>
            <xsd:element name="idx" type="CT_UnsignedInt" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_BandFmts">
        <xsd:sequence>
            <xsd:element name="bandFmt" type="CT_BandFmt" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_SurfaceChartShared">
        <xsd:sequence>
            <xsd:element name="wireframe" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ser" type="CT_SurfaceSer" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="bandFmts" type="CT_BandFmts" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_SurfaceChart">
        <xsd:sequence>
            <xsd:group ref="EG_SurfaceChartShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="axId" type="CT_UnsignedInt" minOccurs="2" maxOccurs="3"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Surface3DChart">
```

```
        <xsd:sequence>
            <xsd:group ref="EG_SurfaceChartShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="axId" type="CT_UnsignedInt" minOccurs="3" maxOccurs="3"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_AxPos">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="b"/>
            <xsd:enumeration value="l"/>
            <xsd:enumeration value="r"/>
            <xsd:enumeration value="t"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_AxPos">
        <xsd:attribute name="val" type="ST_AxPos" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Crosses">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="autoZero"/>
            <xsd:enumeration value="max"/>
            <xsd:enumeration value="min"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Crosses">
        <xsd:attribute name="val" type="ST_Crosses" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_CrossBetween">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="between"/>
            <xsd:enumeration value="midCat"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_CrossBetween">
        <xsd:attribute name="val" type="ST_CrossBetween" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TickMark">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="cross"/>
            <xsd:enumeration value="in"/>
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="out"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TickMark">
        <xsd:attribute name="val" type="ST_TickMark" default="cross"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TickLblPos">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="high"/>
            <xsd:enumeration value="low"/>
            <xsd:enumeration value="nextTo"/>
            <xsd:enumeration value="none"/>
```



```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TickLblPos">
        <xsd:attribute name="val" type="ST_TickLblPos" default="nextTo"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Skip">
        <xsd:restriction base="xsd:unsignedInt">
            <xsd:minInclusive value="1"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Skip">
        <xsd:attribute name="val" type="ST_Skip" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TimeUnit">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="days"/>
            <xsd:enumeration value="months"/>
            <xsd:enumeration value="years"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TimeUnit">
        <xsd:attribute name="val" type="ST_TimeUnit" default="days"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_AxisUnit">
        <xsd:restriction base="xsd:double">
            <xsd:minExclusive value="0"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_AxisUnit">
        <xsd:attribute name="val" type="ST_AxisUnit" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_BuiltInUnit">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="hundreds"/>
            <xsd:enumeration value="thousands"/>
            <xsd:enumeration value="tenThousands"/>
            <xsd:enumeration value="hundredThousands"/>
            <xsd:enumeration value="millions"/>
            <xsd:enumeration value="tenMillions"/>
            <xsd:enumeration value="hundredMillions"/>
            <xsd:enumeration value="billions"/>
            <xsd:enumeration value="trillions"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_BuiltInUnit">
        <xsd:attribute name="val" type="ST_BuiltInUnit" default="thousands"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PictureFormat">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="stretch"/>
            <xsd:enumeration value="stack"/>
            <xsd:enumeration value="stackScale"/>
        </xsd:restriction>
```

```
    </xsd:simpleType>
    <xsd:complexType name="CT_PictureFormat">
        <xsd:attribute name="val" type="ST_PictureFormat" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PictureStackUnit">
        <xsd:restriction base="xsd:double">
            <xsd:minExclusive value="0"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PictureStackUnit">
        <xsd:attribute name="val" type="ST_PictureStackUnit" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PictureOptions">
        <xsd:sequence>
            <xsd:element name="applyToFront" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="applyToSides" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="applyToEnd" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pictureFormat" type="CT_PictureFormat" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pictureStackUnit" type="CT_PictureStackUnit" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DispUnitsLbl">
        <xsd:sequence>
            <xsd:element name="layout" type="CT_Layout" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tx" type="CT_Tx" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="txPr" type="a:CT_TextBody" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DispUnits">
        <xsd:sequence>
            <xsd:choice>
                <xsd:element name="custUnit" type="CT_Double" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="builtInUnit" type="CT_BuiltInUnit" minOccurs="1" maxOccurs="1"/>
            </xsd:choice>
            <xsd:element name="dispUnitsLbl" type="CT_DispUnitsLbl" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_Orientation">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="maxMin"/>
            <xsd:enumeration value="minMax"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Orientation">
        <xsd:attribute name="val" type="ST_Orientation" default="minMax"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_LogBase">
        <xsd:restriction base="xsd:double">
            <xsd:minInclusive value="2"/>
            <xsd:maxInclusive value="1000"/>
```



```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_LogBase">
        <xsd:attribute name="val" type="ST_LogBase" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Scaling">
        <xsd:sequence>
            <xsd:element name="logBase" type="CT_LogBase" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="orientation" type="CT_Orientation" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="max" type="CT_Double" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="min" type="CT_Double" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_LblOffset">
        <xsd:union memberTypes="ST_LblOffsetPercent ST_LblOffsetUShort"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_LblOffsetPercent">
        <xsd:restriction base="xsd:string">
               <xsd:pattern value="0*(([0-9])|([1-9][0-9])|([1-9][0-9][0-9])|1000)%"/>
          </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_LblOffsetUShort">
        <xsd:restriction base="xsd:unsignedShort">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="1000"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_LblOffset">
        <xsd:attribute name="val" type="ST_LblOffset" default="100%"/>
    </xsd:complexType>
    <xsd:group name="EG_AxShared">
        <xsd:sequence>
            <xsd:element name="axId" type="CT_UnsignedInt" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="scaling" type="CT_Scaling" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="delete" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="axPos" type="CT_AxPos" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="majorGridlines" type="CT_ChartLines" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="minorGridlines" type="CT_ChartLines" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="title" type="CT_Title" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="numFmt" type="CT_NumFmt" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="majorTickMark" type="CT_TickMark" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="minorTickMark" type="CT_TickMark" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tickLblPos" type="CT_TickLblPos" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="txPr" type="a:CT_TextBody" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="crossAx" type="CT_UnsignedInt" minOccurs="1" maxOccurs="1"/>
            <xsd:choice minOccurs="0" maxOccurs="1">
                <xsd:element name="crosses" type="CT_Crosses" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="crossesAt" type="CT_Double" minOccurs="1" maxOccurs="1"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:group>
```

```
    <xsd:complexType name="CT_CatAx">
        <xsd:sequence>
            <xsd:group ref="EG_AxShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="auto" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="lblAlgn" type="CT_LblAlgn" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="lblOffset" type="CT_LblOffset" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tickLblSkip" type="CT_Skip" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tickMarkSkip" type="CT_Skip" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="noMultiLvlLbl" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DateAx">
        <xsd:sequence>
            <xsd:group ref="EG_AxShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="auto" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="lblOffset" type="CT_LblOffset" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="baseTimeUnit" type="CT_TimeUnit" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="majorUnit" type="CT_AxisUnit" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="majorTimeUnit" type="CT_TimeUnit" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="minorUnit" type="CT_AxisUnit" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="minorTimeUnit" type="CT_TimeUnit" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SerAx">
        <xsd:sequence>
            <xsd:group ref="EG_AxShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="tickLblSkip" type="CT_Skip" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tickMarkSkip" type="CT_Skip" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ValAx">
        <xsd:sequence>
            <xsd:group ref="EG_AxShared" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="crossBetween" type="CT_CrossBetween" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="majorUnit" type="CT_AxisUnit" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="minorUnit" type="CT_AxisUnit" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dispUnits" type="CT_DispUnits" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_PlotArea">
        <xsd:sequence>
            <xsd:element name="layout" type="CT_Layout" minOccurs="0" maxOccurs="1"/>
            <xsd:choice minOccurs="1" maxOccurs="unbounded">
                <xsd:element name="areaChart" type="CT_AreaChart" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="area3DChart" type="CT_Area3DChart" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="lineChart" type="CT_LineChart" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="line3DChart" type="CT_Line3DChart" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="stockChart" type="CT_StockChart" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="radarChart" type="CT_RadarChart" minOccurs="1" maxOccurs="1"/>
```



```
                <xsd:element name="scatterChart" type="CT_ScatterChart" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="pieChart" type="CT_PieChart" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="pie3DChart" type="CT_Pie3DChart" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="doughnutChart" type="CT_DoughnutChart" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="barChart" type="CT_BarChart" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="bar3DChart" type="CT_Bar3DChart" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="ofPieChart" type="CT_OfPieChart" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="surfaceChart" type="CT_SurfaceChart" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="surface3DChart" type="CT_Surface3DChart" minOccurs="1"
                  maxOccurs="1"/>
                <xsd:element name="bubbleChart" type="CT_BubbleChart" minOccurs="1" maxOccurs="1"/>
            </xsd:choice>
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="valAx" type="CT_ValAx" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="catAx" type="CT_CatAx" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="dateAx" type="CT_DateAx" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="serAx" type="CT_SerAx" minOccurs="1" maxOccurs="1"/>
            </xsd:choice>
            <xsd:element name="dTable" type="CT_DTable" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotFmt">
        <xsd:sequence>
            <xsd:element name="idx" type="CT_UnsignedInt" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="txPr" type="a:CT_TextBody" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="marker" type="CT_Marker" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dLbl" type="CT_DLbl" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotFmts">
        <xsd:sequence>
            <xsd:element name="pivotFmt" type="CT_PivotFmt" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_LegendPos">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="b"/>
            <xsd:enumeration value="tr"/>
            <xsd:enumeration value="l"/>
            <xsd:enumeration value="r"/>
            <xsd:enumeration value="t"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_LegendPos">
        <xsd:attribute name="val" type="ST_LegendPos" default="r"/>
    </xsd:complexType>
    <xsd:group name="EG_LegendEntryData">
        <xsd:sequence>
            <xsd:element name="txPr" type="a:CT_TextBody" minOccurs="0" maxOccurs="1"/>
```

```
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_LegendEntry">
        <xsd:sequence>
            <xsd:element name="idx" type="CT_UnsignedInt" minOccurs="1" maxOccurs="1"/>
            <xsd:choice>
                <xsd:element name="delete" type="CT_Boolean" minOccurs="1" maxOccurs="1"/>
                <xsd:group ref="EG_LegendEntryData" minOccurs="1" maxOccurs="1"/>
            </xsd:choice>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Legend">
        <xsd:sequence>
            <xsd:element name="legendPos" type="CT_LegendPos" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="legendEntry" type="CT_LegendEntry" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="layout" type="CT_Layout" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="overlay" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="txPr" type="a:CT_TextBody" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_DispBlanksAs">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="span"/>
            <xsd:enumeration value="gap"/>
            <xsd:enumeration value="zero"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DispBlanksAs">
        <xsd:attribute name="val" type="ST_DispBlanksAs" default="zero"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Chart">
        <xsd:sequence>
            <xsd:element name="title" type="CT_Title" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="autoTitleDeleted" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pivotFmts" type="CT_PivotFmts" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="view3D" type="CT_View3D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="floor" type="CT_Surface" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sideWall" type="CT_Surface" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="backWall" type="CT_Surface" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="plotArea" type="CT_PlotArea" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="legend" type="CT_Legend" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="plotVisOnly" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dispBlanksAs" type="CT_DispBlanksAs" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="showDLblsOverMax" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_Style">
        <xsd:restriction base="xsd:unsignedByte">
```



```
            <xsd:minInclusive value="1"/>
            <xsd:maxInclusive value="48"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Style">
        <xsd:attribute name="val" type="ST_Style" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotSource">
        <xsd:sequence>
            <xsd:element name="name" type="s:ST_Xstring" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="fmtId" type="CT_UnsignedInt" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Protection">
        <xsd:sequence>
            <xsd:element name="chartObject" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="data" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="formatting" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="selection" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="userInterface" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_HeaderFooter">
        <xsd:sequence>
            <xsd:element name="oddHeader" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="oddFooter" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="evenHeader" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="evenFooter" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="firstHeader" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="firstFooter" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="alignWithMargins" type="xsd:boolean" default="true"/>
        <xsd:attribute name="differentOddEven" type="xsd:boolean" default="false"/>
        <xsd:attribute name="differentFirst" type="xsd:boolean" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PageMargins">
        <xsd:attribute name="l" type="xsd:double" use="required"/>
        <xsd:attribute name="r" type="xsd:double" use="required"/>
        <xsd:attribute name="t" type="xsd:double" use="required"/>
        <xsd:attribute name="b" type="xsd:double" use="required"/>
        <xsd:attribute name="header" type="xsd:double" use="required"/>
        <xsd:attribute name="footer" type="xsd:double" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PageSetupOrientation">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="default"/>
            <xsd:enumeration value="portrait"/>
            <xsd:enumeration value="landscape"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_ExternalData">
        <xsd:sequence>
```



```
            <xsd:element name="autoUpdate" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PageSetup">
        <xsd:attribute name="paperSize" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="paperHeight" type="s:ST_PositiveUniversalMeasure" use="optional"/>
        <xsd:attribute name="paperWidth" type="s:ST_PositiveUniversalMeasure" use="optional"/>
        <xsd:attribute name="firstPageNumber" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="orientation" type="ST_PageSetupOrientation" use="optional"
          default="default"/>
        <xsd:attribute name="blackAndWhite" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="draft" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="useFirstPageNumber" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="horizontalDpi" type="xsd:int" use="optional" default="600"/>
        <xsd:attribute name="verticalDpi" type="xsd:int" use="optional" default="600"/>
        <xsd:attribute name="copies" type="xsd:unsignedInt" use="optional" default="1"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PrintSettings">
        <xsd:sequence>
            <xsd:element name="headerFooter" type="CT_HeaderFooter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pageMargins" type="CT_PageMargins" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pageSetup" type="CT_PageSetup" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="legacyDrawingHF" type="CT_RelId" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ChartSpace">
        <xsd:sequence>
            <xsd:element name="date1904" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="lang" type="CT_TextLanguageID" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="roundedCorners" type="CT_Boolean" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="style" type="CT_Style" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="clrMapOvr" type="a:CT_ColorMapping" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pivotSource" type="CT_PivotSource" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="protection" type="CT_Protection" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="chart" type="CT_Chart" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="txPr" type="a:CT_TextBody" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="externalData" type="CT_ExternalData" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="printSettings" type="CT_PrintSettings" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="userShapes" type="CT_RelId" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="chartSpace" type="CT_ChartSpace"/>
    <xsd:element name="userShapes" type="cdr:CT_Drawing"/>
    <xsd:element name="chart" type="CT_RelId"/>
</xsd:schema>
```

A.5.2 DrawingML - Chart Drawings

This schema is available in the file dml-chartDrawing.xsd.


```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
  xmlns="http://schemas.openxmlformats.org/drawingml/2006/chartDrawing"
  targetNamespace="http://schemas.openxmlformats.org/drawingml/2006/chartDrawing"
  elementFormDefault="qualified">
    <xsd:import namespace="http://schemas.openxmlformats.org/drawingml/2006/main" schemaLocation="dml-
      main.xsd"/>
    <xsd:complexType name="CT_ShapeNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvSpPr" type="a:CT_NonVisualDrawingShapeProps" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Shape">
        <xsd:sequence>
            <xsd:element name="nvSpPr" type="CT_ShapeNonVisual" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="style" type="a:CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="txBody" type="a:CT_TextBody" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="macro" type="xsd:string" use="optional"/>
        <xsd:attribute name="textlink" type="xsd:string" use="optional"/>
        <xsd:attribute name="fLocksText" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="fPublished" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ConnectorNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvCxnSpPr" type="a:CT_NonVisualConnectorProperties" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Connector">
        <xsd:sequence>
            <xsd:element name="nvCxnSpPr" type="CT_ConnectorNonVisual" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="style" type="a:CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="macro" type="xsd:string" use="optional"/>
        <xsd:attribute name="fPublished" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PictureNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvPicPr" type="a:CT_NonVisualPictureProperties" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Picture">
        <xsd:sequence>
            <xsd:element name="nvPicPr" type="CT_PictureNonVisual" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="blipFill" type="a:CT_BlipFillProperties" minOccurs="1" maxOccurs="1"/>
```

```
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="style" type="a:CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="macro" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="fPublished" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GraphicFrameNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvGraphicFramePr" type="a:CT_NonVisualGraphicFrameProperties"
              minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GraphicFrame">
        <xsd:sequence>
            <xsd:element name="nvGraphicFramePr" type="CT_GraphicFrameNonVisual" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="xfrm" type="a:CT_Transform2D" minOccurs="1" maxOccurs="1"/>
            <xsd:element ref="a:graphic" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="macro" type="xsd:string" use="optional"/>
        <xsd:attribute name="fPublished" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupShapeNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvGrpSpPr" type="a:CT_NonVisualGroupDrawingShapeProps" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupShape">
        <xsd:sequence>
            <xsd:element name="nvGrpSpPr" type="CT_GroupShapeNonVisual" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="grpSpPr" type="a:CT_GroupShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="sp" type="CT_Shape"/>
                <xsd:element name="grpSp" type="CT_GroupShape"/>
                <xsd:element name="graphicFrame" type="CT_GraphicFrame"/>
                <xsd:element name="cxnSp" type="CT_Connector"/>
                <xsd:element name="pic" type="CT_Picture"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_ObjectChoices">
        <xsd:sequence>
            <xsd:choice minOccurs="1" maxOccurs="1">
                <xsd:element name="sp" type="CT_Shape"/>
                <xsd:element name="grpSp" type="CT_GroupShape"/>
                <xsd:element name="graphicFrame" type="CT_GraphicFrame"/>
                <xsd:element name="cxnSp" type="CT_Connector"/>
                <xsd:element name="pic" type="CT_Picture"/>
            </xsd:choice>
        </xsd:sequence>
```



```
    </xsd:group>
    <xsd:simpleType name="ST_MarkerCoordinate">
        <xsd:restriction base="xsd:double">
            <xsd:minInclusive value="0.0"/>
            <xsd:maxInclusive value="1.0"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Marker">
        <xsd:sequence>
            <xsd:element name="x" type="ST_MarkerCoordinate" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="y" type="ST_MarkerCoordinate" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_RelSizeAnchor">
        <xsd:sequence>
            <xsd:element name="from" type="CT_Marker"/>
            <xsd:element name="to" type="CT_Marker"/>
            <xsd:group ref="EG_ObjectChoices"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_AbsSizeAnchor">
        <xsd:sequence>
            <xsd:element name="from" type="CT_Marker"/>
            <xsd:element name="ext" type="a:CT_PositiveSize2D"/>
            <xsd:group ref="EG_ObjectChoices"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_Anchor">
        <xsd:choice>
            <xsd:element name="relSizeAnchor" type="CT_RelSizeAnchor"/>
            <xsd:element name="absSizeAnchor" type="CT_AbsSizeAnchor"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_Drawing">
        <xsd:sequence>
            <xsd:group ref="EG_Anchor" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

A.5.3 DrawingML - Diagrams

This schema is available in the file dml-diagram.xsd.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.openxmlformats.org/drawingml/2006/diagram"
  xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
  xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
  xmlns:s="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
  targetNamespace="http://schemas.openxmlformats.org/drawingml/2006/diagram"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
      schemaLocation="shared-relationshipReference.xsd"/>
```

```
    <xsd:import namespace="http://schemas.openxmlformats.org/drawingml/2006/main" schemaLocation="dml-
      main.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
    <xsd:complexType name="CT_CTName">
        <xsd:attribute name="lang" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="val" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CTDescription">
        <xsd:attribute name="lang" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="val" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CTCategory">
        <xsd:attribute name="type" type="xsd:anyURI" use="required"/>
        <xsd:attribute name="pri" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CTCategories">
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="cat" type="CT_CTCategory" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_ClrAppMethod">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="span"/>
            <xsd:enumeration value="cycle"/>
            <xsd:enumeration value="repeat"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_HueDir">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="cw"/>
            <xsd:enumeration value="ccw"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Colors">
        <xsd:sequence>
            <xsd:group ref="a:EG_ColorChoice" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="meth" type="ST_ClrAppMethod" use="optional" default="span"/>
        <xsd:attribute name="hueDir" type="ST_HueDir" use="optional" default="cw"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CTStyleLabel">
        <xsd:sequence>
            <xsd:element name="fillClrLst" type="CT_Colors" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="linClrLst" type="CT_Colors" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="effectClrLst" type="CT_Colors" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="txLinClrLst" type="CT_Colors" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="txFillClrLst" type="CT_Colors" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="txEffectClrLst" type="CT_Colors" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="required"/>
```



```
    </xsd:complexType>
    <xsd:complexType name="CT_ColorTransform">
        <xsd:sequence>
            <xsd:element name="title" type="CT_CTName" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="desc" type="CT_CTDescription" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="catLst" type="CT_CTCategories" minOccurs="0"/>
            <xsd:element name="styleLbl" type="CT_CTStyleLabel" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="uniqueId" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="minVer" type="xsd:string" use="optional" />
    </xsd:complexType>
    <xsd:element name="colorsDef" type="CT_ColorTransform"/>
    <xsd:complexType name="CT_ColorTransformHeader">
        <xsd:sequence>
            <xsd:element name="title" type="CT_CTName" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="desc" type="CT_CTDescription" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="catLst" type="CT_CTCategories" minOccurs="0"/>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="uniqueId" type="xsd:string" use="required"/>
        <xsd:attribute name="minVer" type="xsd:string" use="optional" />
        <xsd:attribute name="resId" type="xsd:int" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:element name="colorsDefHdr" type="CT_ColorTransformHeader"/>
    <xsd:complexType name="CT_ColorTransformHeaderLst">
        <xsd:sequence>
            <xsd:element name="colorsDefHdr" type="CT_ColorTransformHeader" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="colorsDefHdrLst" type="CT_ColorTransformHeaderLst"/>
    <xsd:simpleType name="ST_PtType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="node"/>
            <xsd:enumeration value="asst"/>
            <xsd:enumeration value="doc"/>
            <xsd:enumeration value="pres"/>
            <xsd:enumeration value="parTrans"/>
            <xsd:enumeration value="sibTrans"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Pt">
        <xsd:sequence>
            <xsd:element name="prSet" type="CT_ElemPropSet" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="t" type="a:CT_TextBody" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="modelId" type="ST_ModelId" use="required"/>
```

```
        <xsd:attribute name="type" type="ST_PtType" use="optional" default="node"/>
        <xsd:attribute name="cxnId" type="ST_ModelId" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PtList">
        <xsd:sequence>
            <xsd:element name="pt" type="CT_Pt" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_CxnType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="parOf"/>
            <xsd:enumeration value="presOf"/>
            <xsd:enumeration value="presParOf"/>
            <xsd:enumeration value="unknownRelationship"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Cxn">
        <xsd:sequence>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="modelId" type="ST_ModelId" use="required"/>
        <xsd:attribute name="type" type="ST_CxnType" use="optional" default="parOf"/>
        <xsd:attribute name="srcId" type="ST_ModelId" use="required"/>
        <xsd:attribute name="destId" type="ST_ModelId" use="required"/>
        <xsd:attribute name="srcOrd" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="destOrd" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="parTransId" type="ST_ModelId" use="optional" default="0"/>
        <xsd:attribute name="sibTransId" type="ST_ModelId" use="optional" default="0"/>
        <xsd:attribute name="presId" type="xsd:string" use="optional" default=""/>
    </xsd:complexType>
    <xsd:complexType name="CT_CxnList">
        <xsd:sequence>
            <xsd:element name="cxn" type="CT_Cxn" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DataModel">
        <xsd:sequence>
            <xsd:element name="ptLst" type="CT_PtList"/>
            <xsd:element name="cxnLst" type="CT_CxnList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bg" type="a:CT_BackgroundFormatting" minOccurs="0"/>
            <xsd:element name="whole" type="a:CT_WholeE2oFormatting" minOccurs="0"/>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="dataModel" type="CT_DataModel"/>
    <xsd:attributeGroup name="AG_IteratorAttributes">
        <xsd:attribute name="axis" type="ST_AxisTypes" use="optional" default="none"/>
        <xsd:attribute name="ptType" type="ST_ElementTypes" use="optional" default="all"/>
        <xsd:attribute name="hideLastTrans" type="ST_Booleans" use="optional" default="true"/>
        <xsd:attribute name="st" type="ST_Ints" use="optional" default="1"/>
        <xsd:attribute name="cnt" type="ST_UnsignedInts" use="optional" default="0"/>
```



```
        <xsd:attribute name="step" type="ST_Ints" use="optional" default="1"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="AG_ConstraintAttributes">
        <xsd:attribute name="type" type="ST_ConstraintType" use="required"/>
        <xsd:attribute name="for" type="ST_ConstraintRelationship" use="optional" default="self"/>
        <xsd:attribute name="forName" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="ptType" type="ST_ElementType" use="optional" default="all"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="AG_ConstraintRefAttributes">
        <xsd:attribute name="refType" type="ST_ConstraintType" use="optional" default="none"/>
        <xsd:attribute name="refFor" type="ST_ConstraintRelationship" use="optional" default="self"/>
        <xsd:attribute name="refForName" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="refPtType" type="ST_ElementType" use="optional" default="all"/>
    </xsd:attributeGroup>
    <xsd:complexType name="CT_Constraint">
        <xsd:sequence>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_ConstraintAttributes"/>
        <xsd:attributeGroup ref="AG_ConstraintRefAttributes"/>
        <xsd:attribute name="op" type="ST_BoolOperator" use="optional" default="none"/>
        <xsd:attribute name="val" type="xsd:double" use="optional" default="0"/>
        <xsd:attribute name="fact" type="xsd:double" use="optional" default="1"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Constraints">
        <xsd:sequence>
            <xsd:element name="constr" type="CT_Constraint" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_NumericRule">
        <xsd:sequence>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_ConstraintAttributes"/>
        <xsd:attribute name="val" type="xsd:double" use="optional" default="NaN"/>
        <xsd:attribute name="fact" type="xsd:double" use="optional" default="NaN"/>
        <xsd:attribute name="max" type="xsd:double" use="optional" default="NaN"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Rules">
        <xsd:sequence>
            <xsd:element name="rule" type="CT_NumericRule" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_PresentationOf">
        <xsd:sequence>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_IteratorAttributes"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_LayoutShapeType" final="restriction">
```

```
        <xsd:union memberTypes="a:ST_ShapeType ST_OutputShapeType"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Index1">
        <xsd:restriction base="xsd:unsignedInt">
            <xsd:minInclusive value="1"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Adj">
        <xsd:attribute name="idx" type="ST_Index1" use="required"/>
        <xsd:attribute name="val" type="xsd:double" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AdjLst">
        <xsd:sequence>
            <xsd:element name="adj" type="CT_Adj" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Shape">
        <xsd:sequence>
            <xsd:element name="adjLst" type="CT_AdjLst" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="rot" type="xsd:double" use="optional" default="0"/>
        <xsd:attribute name="type" type="ST_LayoutShapeType" use="optional" default="none"/>
        <xsd:attribute ref="r:blip" use="optional"/>
        <xsd:attribute name="zOrderOff" type="xsd:int" use="optional" default="0"/>
        <xsd:attribute name="hideGeom" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="lkTxEntry" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="blipPhldr" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Parameter">
        <xsd:attribute name="type" type="ST_ParameterId" use="required"/>
        <xsd:attribute name="val" type="ST_ParameterVal" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Algorithm">
        <xsd:sequence>
            <xsd:element name="param" type="CT_Parameter" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="type" type="ST_AlgorithmType" use="required"/>
        <xsd:attribute name="rev" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_LayoutNode">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="alg" type="CT_Algorithm" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="shape" type="CT_Shape" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="presOf" type="CT_PresentationOf" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="constrLst" type="CT_Constraints" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ruleLst" type="CT_Rules" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="varLst" type="CT_LayoutVariablePropertySet" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="forEach" type="CT_ForEach"/>
```



```
            <xsd:element name="layoutNode" type="CT_LayoutNode"/>
            <xsd:element name="choose" type="CT_Choose"/>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:choice>
        <xsd:attribute name="name" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="styleLbl" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="chOrder" type="ST_ChildOrderType" use="optional" default="b"/>
        <xsd:attribute name="moveWith" type="xsd:string" use="optional" default=""/>
    </xsd:complexType>
    <xsd:complexType name="CT_ForEach">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="alg" type="CT_Algorithm" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="shape" type="CT_Shape" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="presOf" type="CT_PresentationOf" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="constrLst" type="CT_Constraints" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ruleLst" type="CT_Rules" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="forEach" type="CT_ForEach"/>
            <xsd:element name="layoutNode" type="CT_LayoutNode"/>
            <xsd:element name="choose" type="CT_Choose"/>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:choice>
        <xsd:attribute name="name" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="ref" type="xsd:string" use="optional" default=""/>
        <xsd:attributeGroup ref="AG_IteratorAttributes"/>
    </xsd:complexType>
    <xsd:complexType name="CT_When">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="alg" type="CT_Algorithm" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="shape" type="CT_Shape" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="presOf" type="CT_PresentationOf" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="constrLst" type="CT_Constraints" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ruleLst" type="CT_Rules" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="forEach" type="CT_ForEach"/>
            <xsd:element name="layoutNode" type="CT_LayoutNode"/>
            <xsd:element name="choose" type="CT_Choose"/>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:choice>
        <xsd:attribute name="name" type="xsd:string" use="optional" default=""/>
        <xsd:attributeGroup ref="AG_IteratorAttributes"/>
        <xsd:attribute name="func" type="ST_FunctionType" use="required"/>
        <xsd:attribute name="arg" type="ST_FunctionArgument" use="optional" default="none"/>
        <xsd:attribute name="op" type="ST_FunctionOperator" use="required"/>
        <xsd:attribute name="val" type="ST_FunctionValue" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Otherwise">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="alg" type="CT_Algorithm" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="shape" type="CT_Shape" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="presOf" type="CT_PresentationOf" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="constrLst" type="CT_Constraints" minOccurs="0" maxOccurs="1"/>
```

```
            <xsd:element name="ruleLst" type="CT_Rules" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="forEach" type="CT_ForEach"/>
            <xsd:element name="layoutNode" type="CT_LayoutNode"/>
            <xsd:element name="choose" type="CT_Choose"/>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:choice>
        <xsd:attribute name="name" type="xsd:string" use="optional" default=""/>
    </xsd:complexType>
    <xsd:complexType name="CT_Choose">
        <xsd:sequence>
            <xsd:element name="if" type="CT_When" maxOccurs="unbounded"/>
            <xsd:element name="else" type="CT_Otherwise" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="optional" default=""/>
    </xsd:complexType>
    <xsd:complexType name="CT_SampleData">
        <xsd:sequence>
            <xsd:element name="dataModel" type="CT_DataModel" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="useDef" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Category">
        <xsd:attribute name="type" type="xsd:anyURI" use="required"/>
        <xsd:attribute name="pri" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Categories">
        <xsd:sequence>
            <xsd:element name="cat" type="CT_Category" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Name">
        <xsd:attribute name="lang" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="val" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Description">
        <xsd:attribute name="lang" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="val" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DiagramDefinition">
        <xsd:sequence>
            <xsd:element name="title" type="CT_Name" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="desc" type="CT_Description" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="catLst" type="CT_Categories" minOccurs="0"/>
            <xsd:element name="sampData" type="CT_SampleData" minOccurs="0"/>
            <xsd:element name="styleData" type="CT_SampleData" minOccurs="0"/>
            <xsd:element name="clrData" type="CT_SampleData" minOccurs="0"/>
            <xsd:element name="layoutNode" type="CT_LayoutNode"/>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="uniqueId" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="minVer" type="xsd:string" use="optional" />
```



```
        <xsd:attribute name="defStyle" type="xsd:string" use="optional" default=""/>
    </xsd:complexType>
    <xsd:element name="layoutDef" type="CT_DiagramDefinition"/>
    <xsd:complexType name="CT_DiagramDefinitionHeader">
        <xsd:sequence>
            <xsd:element name="title" type="CT_Name" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="desc" type="CT_Description" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="catLst" type="CT_Categories" minOccurs="0"/>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="uniqueId" type="xsd:string" use="required"/>
        <xsd:attribute name="minVer" type="xsd:string" use="optional" />
        <xsd:attribute name="defStyle" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="resId" type="xsd:int" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:element name="layoutDefHdr" type="CT_DiagramDefinitionHeader"/>
    <xsd:complexType name="CT_DiagramDefinitionHeaderLst">
        <xsd:sequence>
            <xsd:element name="layoutDefHdr" type="CT_DiagramDefinitionHeader" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="layoutDefHdrLst" type="CT_DiagramDefinitionHeaderLst"/>
    <xsd:complexType name="CT_RelIds">
        <xsd:attribute ref="r:dm" use="required"/>
        <xsd:attribute ref="r:lo" use="required"/>
        <xsd:attribute ref="r:qs" use="required"/>
        <xsd:attribute ref="r:cs" use="required"/>
    </xsd:complexType>
    <xsd:element name="relIds" type="CT_RelIds"/>
    <xsd:simpleType name="ST_ParameterVal">
        <xsd:union memberTypes="ST_DiagramHorizontalAlignment ST_VerticalAlignment ST_ChildDirection
          ST_ChildAlignment ST_SecondaryChildAlignment ST_LinearDirection ST_SecondaryLinearDirection
          ST_StartingElement ST_BendPoint ST_ConnectorRouting ST_ArrowheadStyle ST_ConnectorDimension
          ST_RotationPath ST_CenterShapeMapping ST_NodeHorizontalAlignment ST_NodeVerticalAlignment
          ST_FallbackDimension ST_TextDirection ST_PyramidAccentPosition ST_PyramidAccentTextMargin
          ST_TextBlockDirection ST_TextAnchorHorizontal ST_TextAnchorVertical ST_DiagramTextAlignment
          ST_AutoTextRotation ST_GrowDirection ST_FlowDirection ST_ContinueDirection ST_Breakpoint
          ST_Offset ST_HierarchyAlignment xsd:int xsd:double xsd:boolean xsd:string
          ST_ConnectorPoint"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ModelId">
        <xsd:union memberTypes="xsd:int s:ST_Guid"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PrSetCustVal">
        <xsd:union memberTypes="s:ST_Percentage xsd:int"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_ElemPropSet">
        <xsd:sequence>
            <xsd:element name="presLayoutVars" type="CT_LayoutVariablePropertySet" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="style" type="a:CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
```

```
        </xsd:sequence>
        <xsd:attribute name="presAssocID" type="ST_ModelId" use="optional"/>
        <xsd:attribute name="presName" type="xsd:string" use="optional"/>
        <xsd:attribute name="presStyleLbl" type="xsd:string" use="optional"/>
        <xsd:attribute name="presStyleIdx" type="xsd:int" use="optional"/>
        <xsd:attribute name="presStyleCnt" type="xsd:int" use="optional"/>
        <xsd:attribute name="loTypeId" type="xsd:string" use="optional"/>
        <xsd:attribute name="loCatId" type="xsd:string" use="optional"/>
        <xsd:attribute name="qsTypeId" type="xsd:string" use="optional"/>
        <xsd:attribute name="qsCatId" type="xsd:string" use="optional"/>
        <xsd:attribute name="csTypeId" type="xsd:string" use="optional"/>
        <xsd:attribute name="csCatId" type="xsd:string" use="optional"/>
        <xsd:attribute name="coherent3DOff" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="phldrT" type="xsd:string" use="optional"/>
        <xsd:attribute name="phldr" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="custAng" type="xsd:int" use="optional"/>
        <xsd:attribute name="custFlipVert" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="custFlipHor" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="custSzX" type="xsd:int" use="optional"/>
        <xsd:attribute name="custSzY" type="xsd:int" use="optional"/>
        <xsd:attribute name="custScaleX" type="ST_PrSetCustVal" use="optional"/>
        <xsd:attribute name="custScaleY" type="ST_PrSetCustVal" use="optional"/>
        <xsd:attribute name="custT" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="custLinFactX" type="ST_PrSetCustVal" use="optional"/>
        <xsd:attribute name="custLinFactY" type="ST_PrSetCustVal" use="optional"/>
        <xsd:attribute name="custLinFactNeighborX" type="ST_PrSetCustVal" use="optional"/>
        <xsd:attribute name="custLinFactNeighborY" type="ST_PrSetCustVal" use="optional"/>
        <xsd:attribute name="custRadScaleRad" type="ST_PrSetCustVal" use="optional"/>
        <xsd:attribute name="custRadScaleInc" type="ST_PrSetCustVal" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Direction" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="norm"/>
            <xsd:enumeration value="rev"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_HierBranchStyle" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="l"/>
            <xsd:enumeration value="r"/>
            <xsd:enumeration value="hang"/>
            <xsd:enumeration value="std"/>
            <xsd:enumeration value="init"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_AnimOneStr" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="one"/>
            <xsd:enumeration value="branch"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_AnimLvlStr" final="restriction">
```



```
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="lvl"/>
            <xsd:enumeration value="ctr"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_OrgChart">
        <xsd:attribute name="val" type="xsd:boolean" default="false" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_NodeCount">
        <xsd:restriction base="xsd:int">
            <xsd:minInclusive value="-1"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_ChildMax">
        <xsd:attribute name="val" type="ST_NodeCount" default="-1" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ChildPref">
        <xsd:attribute name="val" type="ST_NodeCount" default="-1" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_BulletEnabled">
        <xsd:attribute name="val" type="xsd:boolean" default="false" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Direction">
        <xsd:attribute name="val" type="ST_Direction" default="norm" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_HierBranchStyle">
        <xsd:attribute name="val" type="ST_HierBranchStyle" default="std" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AnimOne">
        <xsd:attribute name="val" type="ST_AnimOneStr" default="one" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AnimLvl">
        <xsd:attribute name="val" type="ST_AnimLvlStr" default="none" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_ResizeHandlesStr" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="exact"/>
            <xsd:enumeration value="rel"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_ResizeHandles">
        <xsd:attribute name="val" type="ST_ResizeHandlesStr" default="rel" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_LayoutVariablePropertySet">
        <xsd:sequence>
            <xsd:element name="orgChart" type="CT_OrgChart" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="chMax" type="CT_ChildMax" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="chPref" type="CT_ChildPref" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bulletEnabled" type="CT_BulletEnabled" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dir" type="CT_Direction" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="hierBranch" type="CT_HierBranchStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="animOne" type="CT_AnimOne" minOccurs="0" maxOccurs="1"/>
```

```
            <xsd:element name="animLvl" type="CT_AnimLvl" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="resizeHandles" type="CT_ResizeHandles" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SDName">
        <xsd:attribute name="lang" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="val" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SDDescription">
        <xsd:attribute name="lang" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="val" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SDCategory">
        <xsd:attribute name="type" type="xsd:anyURI" use="required"/>
        <xsd:attribute name="pri" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SDCategories">
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="cat" type="CT_SDCategory" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TextProps">
        <xsd:sequence>
            <xsd:group ref="a:EG_Text3D" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_StyleLabel">
        <xsd:sequence>
            <xsd:element name="scene3d" type="a:CT_Scene3D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sp3d" type="a:CT_Shape3D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="txPr" type="CT_TextProps" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="style" type="a:CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_StyleDefinition">
        <xsd:sequence>
            <xsd:element name="title" type="CT_SDName" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="desc" type="CT_SDDescription" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="catLst" type="CT_SDCategories" minOccurs="0"/>
            <xsd:element name="scene3d" type="a:CT_Scene3D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="styleLbl" type="CT_StyleLabel" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="uniqueId" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="minVer" type="xsd:string" use="optional" />
    </xsd:complexType>
    <xsd:element name="styleDef" type="CT_StyleDefinition"/>
    <xsd:complexType name="CT_StyleDefinitionHeader">
        <xsd:sequence>
```



```
            <xsd:element name="title" type="CT_SDName" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="desc" type="CT_SDDescription" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="catLst" type="CT_SDCategories" minOccurs="0"/>
            <xsd:element name="extLst" type="a:CT_OfficeArtExtensionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="uniqueId" type="xsd:string" use="required"/>
        <xsd:attribute name="minVer" type="xsd:string" use="optional" />
        <xsd:attribute name="resId" type="xsd:int" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:element name="styleDefHdr" type="CT_StyleDefinitionHeader"/>
    <xsd:complexType name="CT_StyleDefinitionHeaderLst">
        <xsd:sequence>
            <xsd:element name="styleDefHdr" type="CT_StyleDefinitionHeader" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="styleDefHdrLst" type="CT_StyleDefinitionHeaderLst"/>
    <xsd:simpleType name="ST_AlgorithmType" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="composite"/>
            <xsd:enumeration value="conn"/>
            <xsd:enumeration value="cycle"/>
            <xsd:enumeration value="hierChild"/>
            <xsd:enumeration value="hierRoot"/>
            <xsd:enumeration value="pyra"/>
            <xsd:enumeration value="lin"/>
            <xsd:enumeration value="sp"/>
            <xsd:enumeration value="tx"/>
            <xsd:enumeration value="snake"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_AxisType" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="self"/>
            <xsd:enumeration value="ch"/>
            <xsd:enumeration value="des"/>
            <xsd:enumeration value="desOrSelf"/>
            <xsd:enumeration value="par"/>
            <xsd:enumeration value="ancst"/>
            <xsd:enumeration value="ancstOrSelf"/>
            <xsd:enumeration value="followSib"/>
            <xsd:enumeration value="precedSib"/>
            <xsd:enumeration value="follow"/>
            <xsd:enumeration value="preced"/>
            <xsd:enumeration value="root"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_AxisTypes">
        <xsd:list itemType="ST_AxisType"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_BoolOperator" final="restriction">
```

```
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="equ"/>
            <xsd:enumeration value="gte"/>
            <xsd:enumeration value="lte"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ChildOrderType" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="b"/>
            <xsd:enumeration value="t"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ConstraintType" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="alignOff"/>
            <xsd:enumeration value="begMarg"/>
            <xsd:enumeration value="bendDist"/>
            <xsd:enumeration value="begPad"/>
            <xsd:enumeration value="b"/>
            <xsd:enumeration value="bMarg"/>
            <xsd:enumeration value="bOff"/>
            <xsd:enumeration value="ctrX"/>
            <xsd:enumeration value="ctrXOff"/>
            <xsd:enumeration value="ctrY"/>
            <xsd:enumeration value="ctrYOff"/>
            <xsd:enumeration value="connDist"/>
            <xsd:enumeration value="diam"/>
            <xsd:enumeration value="endMarg"/>
            <xsd:enumeration value="endPad"/>
            <xsd:enumeration value="h"/>
            <xsd:enumeration value="hArH"/>
            <xsd:enumeration value="hOff"/>
            <xsd:enumeration value="l"/>
            <xsd:enumeration value="lMarg"/>
            <xsd:enumeration value="lOff"/>
            <xsd:enumeration value="r"/>
            <xsd:enumeration value="rMarg"/>
            <xsd:enumeration value="rOff"/>
            <xsd:enumeration value="primFontSz"/>
            <xsd:enumeration value="pyraAcctRatio"/>
            <xsd:enumeration value="secFontSz"/>
            <xsd:enumeration value="sibSp"/>
            <xsd:enumeration value="secSibSp"/>
            <xsd:enumeration value="sp"/>
            <xsd:enumeration value="stemThick"/>
            <xsd:enumeration value="t"/>
            <xsd:enumeration value="tMarg"/>
            <xsd:enumeration value="tOff"/>
            <xsd:enumeration value="userA"/>
            <xsd:enumeration value="userB"/>
            <xsd:enumeration value="userC"/>
```



```
            <xsd:enumeration value="userD"/>
            <xsd:enumeration value="userE"/>
            <xsd:enumeration value="userF"/>
            <xsd:enumeration value="userG"/>
            <xsd:enumeration value="userH"/>
            <xsd:enumeration value="userI"/>
            <xsd:enumeration value="userJ"/>
            <xsd:enumeration value="userK"/>
            <xsd:enumeration value="userL"/>
            <xsd:enumeration value="userM"/>
            <xsd:enumeration value="userN"/>
            <xsd:enumeration value="userO"/>
            <xsd:enumeration value="userP"/>
            <xsd:enumeration value="userQ"/>
            <xsd:enumeration value="userR"/>
            <xsd:enumeration value="userS"/>
            <xsd:enumeration value="userT"/>
            <xsd:enumeration value="userU"/>
            <xsd:enumeration value="userV"/>
            <xsd:enumeration value="userW"/>
            <xsd:enumeration value="userX"/>
            <xsd:enumeration value="userY"/>
            <xsd:enumeration value="userZ"/>
            <xsd:enumeration value="w"/>
            <xsd:enumeration value="wArH"/>
            <xsd:enumeration value="wOff"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ConstraintRelationship" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="self"/>
            <xsd:enumeration value="ch"/>
            <xsd:enumeration value="des"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ElementType" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="all"/>
            <xsd:enumeration value="doc"/>
            <xsd:enumeration value="node"/>
            <xsd:enumeration value="norm"/>
            <xsd:enumeration value="nonNorm"/>
            <xsd:enumeration value="asst"/>
            <xsd:enumeration value="nonAsst"/>
            <xsd:enumeration value="parTrans"/>
            <xsd:enumeration value="pres"/>
            <xsd:enumeration value="sibTrans"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ElementTypes">
        <xsd:list itemType="ST_ElementType"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ParameterId" final="restriction">
```

```
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="horzAlign"/>
            <xsd:enumeration value="vertAlign"/>
            <xsd:enumeration value="chDir"/>
            <xsd:enumeration value="chAlign"/>
            <xsd:enumeration value="secChAlign"/>
            <xsd:enumeration value="linDir"/>
            <xsd:enumeration value="secLinDir"/>
            <xsd:enumeration value="stElem"/>
            <xsd:enumeration value="bendPt"/>
            <xsd:enumeration value="connRout"/>
            <xsd:enumeration value="begSty"/>
            <xsd:enumeration value="endSty"/>
            <xsd:enumeration value="dim"/>
            <xsd:enumeration value="rotPath"/>
            <xsd:enumeration value="ctrShpMap"/>
            <xsd:enumeration value="nodeHorzAlign"/>
            <xsd:enumeration value="nodeVertAlign"/>
            <xsd:enumeration value="fallback"/>
            <xsd:enumeration value="txDir"/>
            <xsd:enumeration value="pyraAcctPos"/>
            <xsd:enumeration value="pyraAcctTxMar"/>
            <xsd:enumeration value="txBlDir"/>
            <xsd:enumeration value="txAnchorHorz"/>
            <xsd:enumeration value="txAnchorVert"/>
            <xsd:enumeration value="txAnchorHorzCh"/>
            <xsd:enumeration value="txAnchorVertCh"/>
            <xsd:enumeration value="parTxLTRAlign"/>
            <xsd:enumeration value="parTxRTLAlign"/>
            <xsd:enumeration value="shpTxLTRAlignCh"/>
            <xsd:enumeration value="shpTxRTLAlignCh"/>
            <xsd:enumeration value="autoTxRot"/>
            <xsd:enumeration value="grDir"/>
            <xsd:enumeration value="flowDir"/>
            <xsd:enumeration value="contDir"/>
            <xsd:enumeration value="bkpt"/>
            <xsd:enumeration value="off"/>
            <xsd:enumeration value="hierAlign"/>
            <xsd:enumeration value="bkPtFixedVal"/>
            <xsd:enumeration value="stBulletLvl"/>
            <xsd:enumeration value="stAng"/>
            <xsd:enumeration value="spanAng"/>
            <xsd:enumeration value="ar"/>
            <xsd:enumeration value="lnSpPar"/>
            <xsd:enumeration value="lnSpAfParP"/>
            <xsd:enumeration value="lnSpCh"/>
            <xsd:enumeration value="lnSpAfChP"/>
            <xsd:enumeration value="rtShortDist"/>
            <xsd:enumeration value="alignTx"/>
            <xsd:enumeration value="pyraLvlNode"/>
            <xsd:enumeration value="pyraAcctBkgdNode"/>
            <xsd:enumeration value="pyraAcctTxNode"/>
            <xsd:enumeration value="srcNode"/>
```



```
            <xsd:enumeration value="dstNode"/>
            <xsd:enumeration value="begPts"/>
            <xsd:enumeration value="endPts"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Ints">
        <xsd:list itemType="xsd:int"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_UnsignedInts">
        <xsd:list itemType="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Booleans">
        <xsd:list itemType="xsd:boolean"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FunctionType" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="cnt"/>
            <xsd:enumeration value="pos"/>
            <xsd:enumeration value="revPos"/>
            <xsd:enumeration value="posEven"/>
            <xsd:enumeration value="posOdd"/>
            <xsd:enumeration value="var"/>
            <xsd:enumeration value="depth"/>
            <xsd:enumeration value="maxDepth"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FunctionOperator" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="equ"/>
            <xsd:enumeration value="neq"/>
            <xsd:enumeration value="gt"/>
            <xsd:enumeration value="lt"/>
            <xsd:enumeration value="gte"/>
            <xsd:enumeration value="lte"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_DiagramHorizontalAlignment" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="l"/>
            <xsd:enumeration value="ctr"/>
            <xsd:enumeration value="r"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_VerticalAlignment" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="t"/>
            <xsd:enumeration value="mid"/>
            <xsd:enumeration value="b"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ChildDirection" final="restriction">
```

```
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="horz"/>
            <xsd:enumeration value="vert"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ChildAlignment" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="t"/>
            <xsd:enumeration value="b"/>
            <xsd:enumeration value="l"/>
            <xsd:enumeration value="r"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_SecondaryChildAlignment" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="t"/>
            <xsd:enumeration value="b"/>
            <xsd:enumeration value="l"/>
            <xsd:enumeration value="r"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_LinearDirection" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="fromL"/>
            <xsd:enumeration value="fromR"/>
            <xsd:enumeration value="fromT"/>
            <xsd:enumeration value="fromB"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_SecondaryLinearDirection" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="fromL"/>
            <xsd:enumeration value="fromR"/>
            <xsd:enumeration value="fromT"/>
            <xsd:enumeration value="fromB"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_StartingElement" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="node"/>
            <xsd:enumeration value="trans"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_RotationPath" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="alongPath"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_CenterShapeMapping" final="restriction">
        <xsd:restriction base="xsd:token">
```



```
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="fNode"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_BendPoint" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="beg"/>
            <xsd:enumeration value="def"/>
            <xsd:enumeration value="end"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ConnectorRouting" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="stra"/>
            <xsd:enumeration value="bend"/>
            <xsd:enumeration value="curve"/>
            <xsd:enumeration value="longCurve"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ArrowheadStyle" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="auto"/>
            <xsd:enumeration value="arr"/>
            <xsd:enumeration value="noArr"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ConnectorDimension" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="1D"/>
            <xsd:enumeration value="2D"/>
            <xsd:enumeration value="cust"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ConnectorPoint" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="auto"/>
            <xsd:enumeration value="bCtr"/>
            <xsd:enumeration value="ctr"/>
            <xsd:enumeration value="midL"/>
            <xsd:enumeration value="midR"/>
            <xsd:enumeration value="tCtr"/>
            <xsd:enumeration value="bL"/>
            <xsd:enumeration value="bR"/>
            <xsd:enumeration value="tL"/>
            <xsd:enumeration value="tR"/>
            <xsd:enumeration value="radial"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_NodeHorizontalAlignment" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="l"/>
            <xsd:enumeration value="ctr"/>
            <xsd:enumeration value="r"/>
```

```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_NodeVerticalAlignment" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="t"/>
            <xsd:enumeration value="mid"/>
            <xsd:enumeration value="b"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FallbackDimension" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="1D"/>
            <xsd:enumeration value="2D"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextDirection" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="fromT"/>
            <xsd:enumeration value="fromB"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PyramidAccentPosition" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="bef"/>
            <xsd:enumeration value="aft"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PyramidAccentTextMargin" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="step"/>
            <xsd:enumeration value="stack"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextBlockDirection" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="horz"/>
            <xsd:enumeration value="vert"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextAnchorHorizontal" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="ctr"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextAnchorVertical" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="t"/>
            <xsd:enumeration value="mid"/>
            <xsd:enumeration value="b"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_DiagramTextAlignment" final="restriction">
```



```
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="l"/>
            <xsd:enumeration value="ctr"/>
            <xsd:enumeration value="r"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_AutoTextRotation" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="upr"/>
            <xsd:enumeration value="grav"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_GrowDirection" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="tL"/>
            <xsd:enumeration value="tR"/>
            <xsd:enumeration value="bL"/>
            <xsd:enumeration value="bR"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FlowDirection" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="row"/>
            <xsd:enumeration value="col"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ContinueDirection" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="revDir"/>
            <xsd:enumeration value="sameDir"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Breakpoint" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="endCnv"/>
            <xsd:enumeration value="bal"/>
            <xsd:enumeration value="fixed"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Offset" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="ctr"/>
            <xsd:enumeration value="off"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_HierarchyAlignment" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="tL"/>
            <xsd:enumeration value="tR"/>
            <xsd:enumeration value="tCtrCh"/>
            <xsd:enumeration value="tCtrDes"/>
            <xsd:enumeration value="bL"/>
```

```
            <xsd:enumeration value="bR"/>
            <xsd:enumeration value="bCtrCh"/>
            <xsd:enumeration value="bCtrDes"/>
            <xsd:enumeration value="lT"/>
            <xsd:enumeration value="lB"/>
            <xsd:enumeration value="lCtrCh"/>
            <xsd:enumeration value="lCtrDes"/>
            <xsd:enumeration value="rT"/>
            <xsd:enumeration value="rB"/>
            <xsd:enumeration value="rCtrCh"/>
            <xsd:enumeration value="rCtrDes"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FunctionValue" final="restriction">
        <xsd:union memberTypes="xsd:int xsd:boolean ST_Direction ST_HierBranchStyle ST_AnimOneStr
          ST_AnimLvlStr ST_ResizeHandlesStr"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_VariableType" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="orgChart"/>
            <xsd:enumeration value="chMax"/>
            <xsd:enumeration value="chPref"/>
            <xsd:enumeration value="bulEnabled"/>
            <xsd:enumeration value="dir"/>
            <xsd:enumeration value="hierBranch"/>
            <xsd:enumeration value="animOne"/>
            <xsd:enumeration value="animLvl"/>
            <xsd:enumeration value="resizeHandles"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FunctionArgument" final="restriction">
        <xsd:union memberTypes="ST_VariableType"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_OutputShapeType" final="restriction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="conn"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>
```

A.6 VML

A.6.1 VML

This schema is available in the file vml-main.xsd.

```
<xsd:schema xmlns="urn:schemas-microsoft-com:vml" xmlns:pvml="urn:schemas-microsoft-
  com:office:powerpoint" xmlns:o="urn:schemas-microsoft-com:office:office"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main" xmlns:w10="urn:schemas-
  microsoft-com:office:word"
```



```
  xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships" xmlns:x="urn:schemas-
  microsoft-com:office:excel"
  xmlns:s="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
  targetNamespace="urn:schemas-microsoft-com:vml" elementFormDefault="qualified"
  attributeFormDefault="unqualified">
    <xsd:import namespace="urn:schemas-microsoft-com:office:office" schemaLocation="vml-
      officeDrawing.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/wordprocessingml/2006/main"
      schemaLocation="wml.xsd"/>
    <xsd:import namespace="urn:schemas-microsoft-com:office:word" schemaLocation="vml-
      wordprocessingDrawing.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
      schemaLocation="shared-relationshipReference.xsd"/>
    <xsd:import namespace="urn:schemas-microsoft-com:office:excel" schemaLocation="vml-
      spreadsheetDrawing.xsd"/>
    <xsd:import namespace="urn:schemas-microsoft-com:office:powerpoint" schemaLocation="vml-
      presentationDrawing.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
    <xsd:attributeGroup name="AG_Id">
        <xsd:attribute name="id" type="xsd:string" use="optional"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="AG_Style">
        <xsd:attribute name="style" type="xsd:string" use="optional"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="AG_Type">
        <xsd:attribute name="type" type="xsd:string" use="optional"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="AG_Adj">
        <xsd:attribute name="adj" type="xsd:string" use="optional"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="AG_Path">
        <xsd:attribute name="path" type="xsd:string" use="optional"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="AG_Fill">
        <xsd:attribute name="filled" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="fillcolor" type="s:ST_ColorType" use="optional"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="AG_Chromakey">
        <xsd:attribute name="chromakey" type="s:ST_ColorType" use="optional"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="AG_Ext">
        <xsd:attribute name="ext" form="qualified" type="ST_Ext"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="AG_CoreAttributes">
        <xsd:attributeGroup ref="AG_Id"/>
        <xsd:attributeGroup ref="AG_Style"/>
        <xsd:attribute name="href" type="xsd:string" use="optional"/>
        <xsd:attribute name="target" type="xsd:string" use="optional"/>
        <xsd:attribute name="class" type="xsd:string" use="optional"/>
        <xsd:attribute name="title" type="xsd:string" use="optional"/>
        <xsd:attribute name="alt" type="xsd:string" use="optional"/>
        <xsd:attribute name="coordsize" type="xsd:string" use="optional"/>
```

```
        <xsd:attribute name="coordorigin" type="xsd:string" use="optional"/>
        <xsd:attribute name="wrapcoords" type="xsd:string" use="optional"/>
        <xsd:attribute name="print" type="s:ST_TrueFalse" use="optional"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="AG_ShapeAttributes">
        <xsd:attributeGroup ref="AG_Chromakey"/>
        <xsd:attributeGroup ref="AG_Fill"/>
        <xsd:attribute name="opacity" type="xsd:string" use="optional"/>
        <xsd:attribute name="stroked" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="strokecolor" type="s:ST_ColorType" use="optional"/>
        <xsd:attribute name="strokeweight" type="xsd:string" use="optional"/>
        <xsd:attribute name="insetpen" type="s:ST_TrueFalse" use="optional"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="AG_OfficeCoreAttributes">
        <xsd:attribute ref="o:spid"/>
        <xsd:attribute ref="o:oned"/>
        <xsd:attribute ref="o:regroupid"/>
        <xsd:attribute ref="o:doubleclicknotify"/>
        <xsd:attribute ref="o:button"/>
        <xsd:attribute ref="o:userhidden"/>
        <xsd:attribute ref="o:bullet"/>
        <xsd:attribute ref="o:hr"/>
        <xsd:attribute ref="o:hrstd"/>
        <xsd:attribute ref="o:hrnoshade"/>
        <xsd:attribute ref="o:hrpct"/>
        <xsd:attribute ref="o:hralign"/>
        <xsd:attribute ref="o:allowincell"/>
        <xsd:attribute ref="o:allowoverlap"/>
        <xsd:attribute ref="o:userdrawn"/>
        <xsd:attribute ref="o:bordertopcolor"/>
        <xsd:attribute ref="o:borderleftcolor"/>
        <xsd:attribute ref="o:borderbottomcolor"/>
        <xsd:attribute ref="o:borderrightcolor"/>
        <xsd:attribute ref="o:dgmlayout"/>
        <xsd:attribute ref="o:dgmnodekind"/>
        <xsd:attribute ref="o:dgmlayoutmru"/>
        <xsd:attribute ref="o:insetmode"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="AG_OfficeShapeAttributes">
        <xsd:attribute ref="o:spt"/>
        <xsd:attribute ref="o:connectortype"/>
        <xsd:attribute ref="o:bwmode"/>
        <xsd:attribute ref="o:bwpure"/>
        <xsd:attribute ref="o:bwnormal"/>
        <xsd:attribute ref="o:forcedash"/>
        <xsd:attribute ref="o:oleicon"/>
        <xsd:attribute ref="o:ole"/>
        <xsd:attribute ref="o:preferrelative"/>
        <xsd:attribute ref="o:cliptowrap"/>
        <xsd:attribute ref="o:clip"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="AG_AllCoreAttributes">
        <xsd:attributeGroup ref="AG_CoreAttributes"/>
```



```
        <xsd:attributeGroup ref="AG_OfficeCoreAttributes"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="AG_AllShapeAttributes">
        <xsd:attributeGroup ref="AG_ShapeAttributes"/>
        <xsd:attributeGroup ref="AG_OfficeShapeAttributes"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="AG_ImageAttributes">
        <xsd:attribute name="src" type="xsd:string" use="optional"/>
        <xsd:attribute name="cropleft" type="xsd:string" use="optional"/>
        <xsd:attribute name="croptop" type="xsd:string" use="optional"/>
        <xsd:attribute name="cropright" type="xsd:string" use="optional"/>
        <xsd:attribute name="cropbottom" type="xsd:string" use="optional"/>
        <xsd:attribute name="gain" type="xsd:string" use="optional"/>
        <xsd:attribute name="blacklevel" type="xsd:string" use="optional"/>
        <xsd:attribute name="gamma" type="xsd:string" use="optional"/>
        <xsd:attribute name="grayscale" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="bilevel" type="s:ST_TrueFalse" use="optional"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="AG_StrokeAttributes">
        <xsd:attribute name="on" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="weight" type="xsd:string" use="optional"/>
        <xsd:attribute name="color" type="s:ST_ColorType" use="optional"/>
        <xsd:attribute name="opacity" type="xsd:string" use="optional"/>
        <xsd:attribute name="linestyle" type="ST_StrokeLineStyle" use="optional"/>
        <xsd:attribute name="miterlimit" type="xsd:decimal" use="optional"/>
        <xsd:attribute name="joinstyle" type="ST_StrokeJoinStyle" use="optional"/>
        <xsd:attribute name="endcap" type="ST_StrokeEndCap" use="optional"/>
        <xsd:attribute name="dashstyle" type="xsd:string" use="optional"/>
        <xsd:attribute name="filltype" type="ST_FillType" use="optional"/>
        <xsd:attribute name="src" type="xsd:string" use="optional"/>
        <xsd:attribute name="imageaspect" type="ST_ImageAspect" use="optional"/>
        <xsd:attribute name="imagesize" type="xsd:string" use="optional"/>
        <xsd:attribute name="imagealignshape" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="color2" type="s:ST_ColorType" use="optional"/>
        <xsd:attribute name="startarrow" type="ST_StrokeArrowType" use="optional"/>
        <xsd:attribute name="startarrowwidth" type="ST_StrokeArrowWidth" use="optional"/>
        <xsd:attribute name="startarrowlength" type="ST_StrokeArrowLength" use="optional"/>
        <xsd:attribute name="endarrow" type="ST_StrokeArrowType" use="optional"/>
        <xsd:attribute name="endarrowwidth" type="ST_StrokeArrowWidth" use="optional"/>
        <xsd:attribute name="endarrowlength" type="ST_StrokeArrowLength" use="optional"/>
        <xsd:attribute ref="o:href"/>
        <xsd:attribute ref="o:althref"/>
        <xsd:attribute ref="o:title"/>
        <xsd:attribute ref="o:forcedash"/>
        <xsd:attribute ref="r:id" use="optional"/>
        <xsd:attribute name="insetpen" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute ref="o:relid"/>
    </xsd:attributeGroup>
    <xsd:group name="EG_ShapeElements">
        <xsd:choice>
            <xsd:element ref="path"/>
            <xsd:element ref="formulas"/>
            <xsd:element ref="handles"/>
```

```
            <xsd:element ref="fill"/>
            <xsd:element ref="stroke"/>
            <xsd:element ref="shadow"/>
            <xsd:element ref="textbox"/>
            <xsd:element ref="textpath"/>
            <xsd:element ref="imagedata"/>
            <xsd:element ref="o:skew"/>
            <xsd:element ref="o:extrusion"/>
            <xsd:element ref="o:callout"/>
            <xsd:element ref="o:lock"/>
            <xsd:element ref="o:clippath"/>
            <xsd:element ref="o:signatureline"/>
            <xsd:element ref="w10:wrap"/>
            <xsd:element ref="w10:anchorlock"/>
            <xsd:element ref="w10:bordertop"/>
            <xsd:element ref="w10:borderbottom"/>
            <xsd:element ref="w10:borderleft"/>
            <xsd:element ref="w10:borderright"/>
            <xsd:element ref="x:ClientData" minOccurs="0"/>
            <xsd:element ref="pvml:textdata" minOccurs="0"/>
        </xsd:choice>
    </xsd:group>
    <xsd:element name="shape" type="CT_Shape"/>
    <xsd:element name="shapetype" type="CT_Shapetype"/>
    <xsd:element name="group" type="CT_Group"/>
    <xsd:element name="background" type="CT_Background"/>
    <xsd:complexType name="CT_Shape">
        <xsd:choice maxOccurs="unbounded">
            <xsd:group ref="EG_ShapeElements"/>
            <xsd:element ref="o:ink"/>
            <xsd:element ref="pvml:iscomment"/>
            <xsd:element ref="o:equationxml"/>
        </xsd:choice>
        <xsd:attributeGroup ref="AG_AllCoreAttributes"/>
        <xsd:attributeGroup ref="AG_AllShapeAttributes"/>
        <xsd:attributeGroup ref="AG_Type"/>
        <xsd:attributeGroup ref="AG_Adj"/>
        <xsd:attributeGroup ref="AG_Path"/>
        <xsd:attribute ref="o:gfxdata"/>
        <xsd:attribute name="equationxml" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Shapetype">
        <xsd:sequence>
            <xsd:group ref="EG_ShapeElements" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element ref="o:complex" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_AllCoreAttributes"/>
        <xsd:attributeGroup ref="AG_AllShapeAttributes"/>
        <xsd:attributeGroup ref="AG_Adj"/>
        <xsd:attributeGroup ref="AG_Path"/>
        <xsd:attribute ref="o:master"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Group">
```



```
        <xsd:choice maxOccurs="unbounded">
            <xsd:group ref="EG_ShapeElements"/>
            <xsd:element ref="group"/>
            <xsd:element ref="shape"/>
            <xsd:element ref="shapetype"/>
            <xsd:element ref="arc"/>
            <xsd:element ref="curve"/>
            <xsd:element ref="image"/>
            <xsd:element ref="line"/>
            <xsd:element ref="oval"/>
            <xsd:element ref="polyline"/>
            <xsd:element ref="rect"/>
            <xsd:element ref="roundrect"/>
            <xsd:element ref="o:diagram"/>
        </xsd:choice>
        <xsd:attributeGroup ref="AG_AllCoreAttributes"/>
        <xsd:attributeGroup ref="AG_Fill"/>
        <xsd:attribute name="editas" type="ST_EditAs" use="optional"/>
        <xsd:attribute ref="o:tableproperties"/>
        <xsd:attribute ref="o:tablelimits"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Background">
        <xsd:sequence>
            <xsd:element ref="fill" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_Id"/>
        <xsd:attributeGroup ref="AG_Fill"/>
        <xsd:attribute ref="o:bwmode"/>
        <xsd:attribute ref="o:bwpure"/>
        <xsd:attribute ref="o:bwnormal"/>
        <xsd:attribute ref="o:targetscreensize"/>
    </xsd:complexType>
    <xsd:element name="fill" type="CT_Fill"/>
    <xsd:element name="formulas" type="CT_Formulas"/>
    <xsd:element name="handles" type="CT_Handles"/>
    <xsd:element name="imagedata" type="CT_ImageData"/>
    <xsd:element name="path" type="CT_Path"/>
    <xsd:element name="textbox" type="CT_Textbox"/>
    <xsd:element name="shadow" type="CT_Shadow"/>
    <xsd:element name="stroke" type="CT_Stroke"/>
    <xsd:element name="textpath" type="CT_TextPath"/>
    <xsd:complexType name="CT_Fill">
        <xsd:sequence>
            <xsd:element ref="o:fill" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_Id"/>
        <xsd:attribute name="type" type="ST_FillType" use="optional"/>
        <xsd:attribute name="on" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="color" type="s:ST_ColorType" use="optional"/>
        <xsd:attribute name="opacity" type="xsd:string" use="optional"/>
        <xsd:attribute name="color2" type="s:ST_ColorType" use="optional"/>
        <xsd:attribute name="src" type="xsd:string" use="optional"/>
        <xsd:attribute ref="o:href"/>
```

```
        <xsd:attribute ref="o:althref"/>
        <xsd:attribute name="size" type="xsd:string" use="optional"/>
        <xsd:attribute name="origin" type="xsd:string" use="optional"/>
        <xsd:attribute name="position" type="xsd:string" use="optional"/>
        <xsd:attribute name="aspect" type="ST_ImageAspect" use="optional"/>
        <xsd:attribute name="colors" type="xsd:string" use="optional"/>
        <xsd:attribute name="angle" type="xsd:decimal" use="optional"/>
        <xsd:attribute name="alignshape" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="focus" type="xsd:string" use="optional"/>
        <xsd:attribute name="focussize" type="xsd:string" use="optional"/>
        <xsd:attribute name="focusposition" type="xsd:string" use="optional"/>
        <xsd:attribute name="method" type="ST_FillMethod" use="optional"/>
        <xsd:attribute ref="o:detectmouseclick"/>
        <xsd:attribute ref="o:title"/>
        <xsd:attribute ref="o:opacity2"/>
        <xsd:attribute name="recolor" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="rotate" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute ref="r:id" use="optional"/>
        <xsd:attribute ref="o:relid" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Formulas">
        <xsd:sequence>
            <xsd:element name="f" type="CT_F" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_F">
        <xsd:attribute name="eqn" type="xsd:string"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Handles">
        <xsd:sequence>
            <xsd:element name="h" type="CT_H" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_H">
        <xsd:attribute name="position" type="xsd:string"/>
        <xsd:attribute name="polar" type="xsd:string"/>
        <xsd:attribute name="map" type="xsd:string"/>
        <xsd:attribute name="invx" type="s:ST_TrueFalse"/>
        <xsd:attribute name="invy" type="s:ST_TrueFalse"/>
        <xsd:attribute name="switch" type="s:ST_TrueFalseBlank"/>
        <xsd:attribute name="xrange" type="xsd:string"/>
        <xsd:attribute name="yrange" type="xsd:string"/>
        <xsd:attribute name="radiusrange" type="xsd:string"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ImageData">
        <xsd:attributeGroup ref="AG_Id"/>
        <xsd:attributeGroup ref="AG_ImageAttributes"/>
        <xsd:attributeGroup ref="AG_Chromakey"/>
        <xsd:attribute name="embosscolor" type="s:ST_ColorType" use="optional"/>
        <xsd:attribute name="recolortarget" type="s:ST_ColorType"/>
        <xsd:attribute ref="o:href"/>
        <xsd:attribute ref="o:althref"/>
        <xsd:attribute ref="o:title"/>
```



```
        <xsd:attribute ref="o:oleid"/>
        <xsd:attribute ref="o:detectmouseclick"/>
        <xsd:attribute ref="o:movie"/>
        <xsd:attribute ref="o:relid"/>
        <xsd:attribute ref="r:id"/>
        <xsd:attribute ref="r:pict"/>
        <xsd:attribute ref="r:href"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Path">
        <xsd:attributeGroup ref="AG_Id"/>
        <xsd:attribute name="v" type="xsd:string" use="optional"/>
        <xsd:attribute name="limo" type="xsd:string" use="optional"/>
        <xsd:attribute name="textboxrect" type="xsd:string" use="optional"/>
        <xsd:attribute name="fillok" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="strokeok" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="shadowok" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="arrowok" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="gradientshapeok" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="textpathok" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="insetpenok" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute ref="o:connecttype"/>
        <xsd:attribute ref="o:connectlocs"/>
        <xsd:attribute ref="o:connectangles"/>
        <xsd:attribute ref="o:extrusionok"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Shadow">
        <xsd:attributeGroup ref="AG_Id"/>
        <xsd:attribute name="on" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="type" type="ST_ShadowType" use="optional"/>
        <xsd:attribute name="obscured" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="color" type="s:ST_ColorType" use="optional"/>
        <xsd:attribute name="opacity" type="xsd:string" use="optional"/>
        <xsd:attribute name="offset" type="xsd:string" use="optional"/>
        <xsd:attribute name="color2" type="s:ST_ColorType" use="optional"/>
        <xsd:attribute name="offset2" type="xsd:string" use="optional"/>
        <xsd:attribute name="origin" type="xsd:string" use="optional"/>
        <xsd:attribute name="matrix" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Stroke">
        <xsd:sequence>
            <xsd:element ref="o:left" minOccurs="0"/>
            <xsd:element ref="o:top" minOccurs="0"/>
            <xsd:element ref="o:right" minOccurs="0"/>
            <xsd:element ref="o:bottom" minOccurs="0"/>
            <xsd:element ref="o:column" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_Id"/>
        <xsd:attributeGroup ref="AG_StrokeAttributes"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Textbox">
        <xsd:choice>
            <xsd:element ref="w:txbxContent" minOccurs="0"/>
            <xsd:any namespace="##local" processContents="skip"/>
```

```
        </xsd:choice>
        <xsd:attributeGroup ref="AG_Id"/>
        <xsd:attributeGroup ref="AG_Style"/>
        <xsd:attribute name="inset" type="xsd:string" use="optional"/>
        <xsd:attribute ref="o:singleclick"/>
        <xsd:attribute ref="o:insetmode"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TextPath">
        <xsd:attributeGroup ref="AG_Id"/>
        <xsd:attributeGroup ref="AG_Style"/>
        <xsd:attribute name="on" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="fitshape" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="fitpath" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="trim" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="xscale" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="string" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:element name="arc" type="CT_Arc"/>
    <xsd:element name="curve" type="CT_Curve"/>
    <xsd:element name="image" type="CT_Image"/>
    <xsd:element name="line" type="CT_Line"/>
    <xsd:element name="oval" type="CT_Oval"/>
    <xsd:element name="polyline" type="CT_PolyLine"/>
    <xsd:element name="rect" type="CT_Rect"/>
    <xsd:element name="roundrect" type="CT_RoundRect"/>
    <xsd:complexType name="CT_Arc">
        <xsd:sequence>
            <xsd:group ref="EG_ShapeElements" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_AllCoreAttributes"/>
        <xsd:attributeGroup ref="AG_AllShapeAttributes"/>
        <xsd:attribute name="startAngle" type="xsd:decimal" use="optional"/>
        <xsd:attribute name="endAngle" type="xsd:decimal" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Curve">
        <xsd:sequence>
            <xsd:group ref="EG_ShapeElements" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_AllCoreAttributes"/>
        <xsd:attributeGroup ref="AG_AllShapeAttributes"/>
        <xsd:attribute name="from" type="xsd:string" use="optional"/>
        <xsd:attribute name="control1" type="xsd:string" use="optional"/>
        <xsd:attribute name="control2" type="xsd:string" use="optional"/>
        <xsd:attribute name="to" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Image">
        <xsd:sequence>
            <xsd:group ref="EG_ShapeElements" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_AllCoreAttributes"/>
        <xsd:attributeGroup ref="AG_AllShapeAttributes"/>
        <xsd:attributeGroup ref="AG_ImageAttributes"/>
    </xsd:complexType>
```



```
    <xsd:complexType name="CT_Line">
        <xsd:sequence>
            <xsd:group ref="EG_ShapeElements" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_AllCoreAttributes"/>
        <xsd:attributeGroup ref="AG_AllShapeAttributes"/>
        <xsd:attribute name="from" type="xsd:string" use="optional"/>
        <xsd:attribute name="to" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Oval">
        <xsd:choice maxOccurs="unbounded">
            <xsd:group ref="EG_ShapeElements" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
        <xsd:attributeGroup ref="AG_AllCoreAttributes"/>
        <xsd:attributeGroup ref="AG_AllShapeAttributes"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PolyLine">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:group ref="EG_ShapeElements"/>
            <xsd:element ref="o:ink"/>
        </xsd:choice>
        <xsd:attributeGroup ref="AG_AllCoreAttributes"/>
        <xsd:attributeGroup ref="AG_AllShapeAttributes"/>
        <xsd:attribute name="points" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Rect">
        <xsd:choice maxOccurs="unbounded">
            <xsd:group ref="EG_ShapeElements" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
        <xsd:attributeGroup ref="AG_AllCoreAttributes"/>
        <xsd:attributeGroup ref="AG_AllShapeAttributes"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RoundRect">
        <xsd:choice maxOccurs="unbounded">
            <xsd:group ref="EG_ShapeElements" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
        <xsd:attributeGroup ref="AG_AllCoreAttributes"/>
        <xsd:attributeGroup ref="AG_AllShapeAttributes"/>
        <xsd:attribute name="arcsize" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Ext">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="view"/>
            <xsd:enumeration value="edit"/>
            <xsd:enumeration value="backwardCompatible"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FillType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="solid"/>
            <xsd:enumeration value="gradient"/>
            <xsd:enumeration value="gradientRadial"/>
            <xsd:enumeration value="tile"/>
```

```
            <xsd:enumeration value="pattern"/>
            <xsd:enumeration value="frame"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FillMethod">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="linear"/>
            <xsd:enumeration value="sigma"/>
            <xsd:enumeration value="any"/>
            <xsd:enumeration value="linear sigma"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ShadowType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="single"/>
            <xsd:enumeration value="double"/>
            <xsd:enumeration value="emboss"/>
            <xsd:enumeration value="perspective"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_StrokeLineStyle">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="single"/>
            <xsd:enumeration value="thinThin"/>
            <xsd:enumeration value="thinThick"/>
            <xsd:enumeration value="thickThin"/>
            <xsd:enumeration value="thickBetweenThin"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_StrokeJoinStyle">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="round"/>
            <xsd:enumeration value="bevel"/>
            <xsd:enumeration value="miter"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_StrokeEndCap">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="flat"/>
            <xsd:enumeration value="square"/>
            <xsd:enumeration value="round"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_StrokeArrowLength">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="short"/>
            <xsd:enumeration value="medium"/>
            <xsd:enumeration value="long"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_StrokeArrowWidth">
        <xsd:restriction base="xsd:string">
```



```
            <xsd:enumeration value="narrow"/>
            <xsd:enumeration value="medium"/>
            <xsd:enumeration value="wide"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_StrokeArrowType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="block"/>
            <xsd:enumeration value="classic"/>
            <xsd:enumeration value="oval"/>
            <xsd:enumeration value="diamond"/>
            <xsd:enumeration value="open"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ImageAspect">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="ignore"/>
            <xsd:enumeration value="atMost"/>
            <xsd:enumeration value="atLeast"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_EditAs">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="canvas"/>
            <xsd:enumeration value="orgchart"/>
            <xsd:enumeration value="radial"/>
            <xsd:enumeration value="cycle"/>
            <xsd:enumeration value="stacked"/>
            <xsd:enumeration value="venn"/>
            <xsd:enumeration value="bullseye"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>
```

A.6.2 VML - Office Drawing

This schema is available in the file vml-officeDrawing.xsd.

```
<xsd:schema xmlns="urn:schemas-microsoft-com:office:office" xmlns:v="urn:schemas-microsoft-com:vml"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
  xmlns:s="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
  targetNamespace="urn:schemas-microsoft-com:office:office" elementFormDefault="qualified"
  attributeFormDefault="unqualified">
    <xsd:import namespace="urn:schemas-microsoft-com:vml" schemaLocation="vml-main.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
      schemaLocation="shared-relationshipReference.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
    <xsd:attribute name="bwmode" type="ST_BWMode"/>
    <xsd:attribute name="bwpure" type="ST_BWMode"/>
    <xsd:attribute name="bwnormal" type="ST_BWMode"/>
```

```
    <xsd:attribute name="targetscreensize" type="ST_ScreenSize"/>
    <xsd:attribute name="insetmode" type="ST_InsetMode" default="custom"/>
    <xsd:attribute name="spt" type="xsd:float"/>
    <xsd:attribute name="wrapcoords" type="xsd:string"/>
    <xsd:attribute name="oned" type="s:ST_TrueFalse"/>
    <xsd:attribute name="regroupid" type="xsd:integer"/>
    <xsd:attribute name="doubleclicknotify" type="s:ST_TrueFalse"/>
    <xsd:attribute name="connectortype" type="ST_ConnectorType" default="straight"/>
    <xsd:attribute name="button" type="s:ST_TrueFalse"/>
    <xsd:attribute name="userhidden" type="s:ST_TrueFalse"/>
    <xsd:attribute name="forcedash" type="s:ST_TrueFalse"/>
    <xsd:attribute name="oleicon" type="s:ST_TrueFalse"/>
    <xsd:attribute name="ole" type="s:ST_TrueFalseBlank"/>
    <xsd:attribute name="preferrelative" type="s:ST_TrueFalse"/>
    <xsd:attribute name="cliptowrap" type="s:ST_TrueFalse"/>
    <xsd:attribute name="clip" type="s:ST_TrueFalse"/>
    <xsd:attribute name="bullet" type="s:ST_TrueFalse"/>
    <xsd:attribute name="hr" type="s:ST_TrueFalse"/>
    <xsd:attribute name="hrstd" type="s:ST_TrueFalse"/>
    <xsd:attribute name="hrnoshade" type="s:ST_TrueFalse"/>
    <xsd:attribute name="hrpct" type="xsd:float"/>
    <xsd:attribute name="hralign" type="ST_HrAlign" default="left"/>
    <xsd:attribute name="allowincell" type="s:ST_TrueFalse"/>
    <xsd:attribute name="allowoverlap" type="s:ST_TrueFalse"/>
    <xsd:attribute name="userdrawn" type="s:ST_TrueFalse"/>
    <xsd:attribute name="bordertopcolor" type="xsd:string"/>
    <xsd:attribute name="borderleftcolor" type="xsd:string"/>
    <xsd:attribute name="borderbottomcolor" type="xsd:string"/>
    <xsd:attribute name="borderrightcolor" type="xsd:string"/>
    <xsd:attribute name="connecttype" type="ST_ConnectType"/>
    <xsd:attribute name="connectlocs" type="xsd:string"/>
    <xsd:attribute name="connectangles" type="xsd:string"/>
    <xsd:attribute name="master" type="xsd:string"/>
    <xsd:attribute name="extrusionok" type="s:ST_TrueFalse"/>
    <xsd:attribute name="href" type="xsd:string"/>
    <xsd:attribute name="althref" type="xsd:string"/>
    <xsd:attribute name="title" type="xsd:string"/>
    <xsd:attribute name="singleclick" type="s:ST_TrueFalse"/>
    <xsd:attribute name="oleid" type="xsd:float"/>
    <xsd:attribute name="detectmouseclick" type="s:ST_TrueFalse"/>
    <xsd:attribute name="movie" type="xsd:float"/>
    <xsd:attribute name="spid" type="xsd:string"/>
    <xsd:attribute name="opacity2" type="xsd:string"/>
    <xsd:attribute name="relid" type="r:ST_RelationshipId"/>
    <xsd:attribute name="dgmlayout" type="ST_DiagramLayout"/>
    <xsd:attribute name="dgmnodekind" type="xsd:integer"/>
    <xsd:attribute name="dgmlayoutmru" type="ST_DiagramLayout"/>
    <xsd:attribute name="gfxdata" type="xsd:base64Binary"/>
    <xsd:attribute name="tableproperties" type="xsd:string"/>
    <xsd:attribute name="tablelimits" type="xsd:string"/>
    <xsd:element name="shapedefaults" type="CT_ShapeDefaults"/>
    <xsd:element name="shapelayout" type="CT_ShapeLayout"/>
    <xsd:element name="signatureline" type="CT_SignatureLine"/>
```



```
    <xsd:element name="ink" type="CT_Ink"/>
    <xsd:element name="diagram" type="CT_Diagram"/>
    <xsd:element name="equationxml" type="CT_EquationXml"/>
    <xsd:complexType name="CT_ShapeDefaults">
        <xsd:all minOccurs="0">
            <xsd:element ref="v:fill" minOccurs="0"/>
            <xsd:element ref="v:stroke" minOccurs="0"/>
            <xsd:element ref="v:textbox" minOccurs="0"/>
            <xsd:element ref="v:shadow" minOccurs="0"/>
            <xsd:element ref="skew" minOccurs="0"/>
            <xsd:element ref="extrusion" minOccurs="0"/>
            <xsd:element ref="callout" minOccurs="0"/>
            <xsd:element ref="lock" minOccurs="0"/>
            <xsd:element name="colormru" minOccurs="0" type="CT_ColorMru"/>
            <xsd:element name="colormenu" minOccurs="0" type="CT_ColorMenu"/>
        </xsd:all>
        <xsd:attributeGroup ref="v:AG_Ext"/>
        <xsd:attribute name="spidmax" type="xsd:integer" use="optional"/>
        <xsd:attribute name="style" type="xsd:string" use="optional"/>
        <xsd:attribute name="fill" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="fillcolor" type="s:ST_ColorType" use="optional"/>
        <xsd:attribute name="stroke" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="strokecolor" type="s:ST_ColorType"/>
        <xsd:attribute name="allowincell" form="qualified" type="s:ST_TrueFalse"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Ink">
        <xsd:sequence/>
        <xsd:attribute name="i" type="xsd:string"/>
        <xsd:attribute name="annotation" type="s:ST_TrueFalse"/>
        <xsd:attribute name="contentType" type="ST_ContentType" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SignatureLine">
        <xsd:attributeGroup ref="v:AG_Ext"/>
        <xsd:attribute name="issignatureline" type="s:ST_TrueFalse"/>
        <xsd:attribute name="id" type="s:ST_Guid"/>
        <xsd:attribute name="provid" type="s:ST_Guid"/>
        <xsd:attribute name="signinginstructionsset" type="s:ST_TrueFalse"/>
        <xsd:attribute name="allowcomments" type="s:ST_TrueFalse"/>
        <xsd:attribute name="showsigndate" type="s:ST_TrueFalse"/>
        <xsd:attribute name="suggestedsigner" type="xsd:string" form="qualified"/>
        <xsd:attribute name="suggestedsigner2" type="xsd:string" form="qualified"/>
        <xsd:attribute name="suggestedsigneremail" type="xsd:string" form="qualified"/>
        <xsd:attribute name="signinginstructions" type="xsd:string"/>
        <xsd:attribute name="addlxml" type="xsd:string"/>
        <xsd:attribute name="sigprovurl" type="xsd:string"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ShapeLayout">
        <xsd:all>
            <xsd:element name="idmap" type="CT_IdMap" minOccurs="0"/>
            <xsd:element name="regrouptable" type="CT_RegroupTable" minOccurs="0"/>
            <xsd:element name="rules" type="CT_Rules" minOccurs="0"/>
        </xsd:all>
        <xsd:attributeGroup ref="v:AG_Ext"/>
```

```
    </xsd:complexType>
    <xsd:complexType name="CT_IdMap">
        <xsd:attributeGroup ref="v:AG_Ext"/>
        <xsd:attribute name="data" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RegroupTable">
        <xsd:sequence>
            <xsd:element name="entry" type="CT_Entry" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="v:AG_Ext"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Entry">
        <xsd:attribute name="new" type="xsd:int" use="optional"/>
        <xsd:attribute name="old" type="xsd:int" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Rules">
        <xsd:sequence>
            <xsd:element name="r" type="CT_R" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="v:AG_Ext"/>
    </xsd:complexType>
    <xsd:complexType name="CT_R">
        <xsd:sequence>
            <xsd:element name="proxy" type="CT_Proxy" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:string" use="required"/>
        <xsd:attribute name="type" type="ST_RType" use="optional"/>
        <xsd:attribute name="how" type="ST_How" use="optional"/>
        <xsd:attribute name="idref" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Proxy">
        <xsd:attribute name="start" type="s:ST_TrueFalseBlank" use="optional" default="false"/>
        <xsd:attribute name="end" type="s:ST_TrueFalseBlank" use="optional" default="false"/>
        <xsd:attribute name="idref" type="xsd:string" use="optional"/>
        <xsd:attribute name="connectloc" type="xsd:int" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Diagram">
        <xsd:sequence>
            <xsd:element name="relationtable" type="CT_RelationTable" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="v:AG_Ext"/>
        <xsd:attribute name="dgmstyle" type="xsd:integer" use="optional"/>
        <xsd:attribute name="autoformat" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="reverse" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="autolayout" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="dgmscalex" type="xsd:integer" use="optional"/>
        <xsd:attribute name="dgmscaley" type="xsd:integer" use="optional"/>
        <xsd:attribute name="dgmfontsize" type="xsd:integer" use="optional"/>
        <xsd:attribute name="constrainbounds" type="xsd:string" use="optional"/>
        <xsd:attribute name="dgmbasetextscale" type="xsd:integer" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_EquationXml">
        <xsd:sequence>
```



```
            <xsd:any namespace="##any"/>
        </xsd:sequence>
        <xsd:attribute name="contentType" type="ST_AlternateMathContentType" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_AlternateMathContentType">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_RelationTable">
        <xsd:sequence>
            <xsd:element name="rel" type="CT_Relation" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="v:AG_Ext"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Relation">
        <xsd:attributeGroup ref="v:AG_Ext"/>
        <xsd:attribute name="idsrc" type="xsd:string" use="optional"/>
        <xsd:attribute name="iddest" type="xsd:string" use="optional"/>
        <xsd:attribute name="idcntr" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ColorMru">
        <xsd:attributeGroup ref="v:AG_Ext"/>
        <xsd:attribute name="colors" type="xsd:string"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ColorMenu">
        <xsd:attributeGroup ref="v:AG_Ext"/>
        <xsd:attribute name="strokecolor" type="s:ST_ColorType"/>
        <xsd:attribute name="fillcolor" type="s:ST_ColorType"/>
        <xsd:attribute name="shadowcolor" type="s:ST_ColorType"/>
        <xsd:attribute name="extrusioncolor" type="s:ST_ColorType"/>
    </xsd:complexType>
    <xsd:element name="skew" type="CT_Skew"/>
    <xsd:element name="extrusion" type="CT_Extrusion"/>
    <xsd:element name="callout" type="CT_Callout"/>
    <xsd:element name="lock" type="CT_Lock"/>
    <xsd:element name="OLEObject" type="CT_OLEObject"/>
    <xsd:element name="complex" type="CT_Complex"/>
    <xsd:element name="left" type="CT_StrokeChild"/>
    <xsd:element name="top" type="CT_StrokeChild"/>
    <xsd:element name="right" type="CT_StrokeChild"/>
    <xsd:element name="bottom" type="CT_StrokeChild"/>
    <xsd:element name="column" type="CT_StrokeChild"/>
    <xsd:element name="clippath" type="CT_ClipPath"/>
    <xsd:element name="fill" type="CT_Fill"/>
    <xsd:complexType name="CT_Skew">
        <xsd:attributeGroup ref="v:AG_Ext"/>
        <xsd:attribute name="id" type="xsd:string" use="optional"/>
        <xsd:attribute name="on" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="offset" type="xsd:string" use="optional"/>
        <xsd:attribute name="origin" type="xsd:string" use="optional"/>
        <xsd:attribute name="matrix" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Extrusion">
        <xsd:attributeGroup ref="v:AG_Ext"/>
```

```
        <xsd:attribute name="on" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="type" type="ST_ExtrusionType" default="parallel" use="optional"/>
        <xsd:attribute name="render" type="ST_ExtrusionRender" default="solid" use="optional"/>
        <xsd:attribute name="viewpointorigin" type="xsd:string" use="optional"/>
        <xsd:attribute name="viewpoint" type="xsd:string" use="optional"/>
        <xsd:attribute name="plane" type="ST_ExtrusionPlane" default="XY" use="optional"/>
        <xsd:attribute name="skewangle" type="xsd:float" use="optional"/>
        <xsd:attribute name="skewamt" type="xsd:string" use="optional"/>
        <xsd:attribute name="foredepth" type="xsd:string" use="optional"/>
        <xsd:attribute name="backdepth" type="xsd:string" use="optional"/>
        <xsd:attribute name="orientation" type="xsd:string" use="optional"/>
        <xsd:attribute name="orientationangle" type="xsd:float" use="optional"/>
        <xsd:attribute name="lockrotationcenter" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="autorotationcenter" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="rotationcenter" type="xsd:string" use="optional"/>
        <xsd:attribute name="rotationangle" type="xsd:string" use="optional"/>
        <xsd:attribute name="colormode" type="ST_ColorMode" use="optional"/>
        <xsd:attribute name="color" type="s:ST_ColorType" use="optional"/>
        <xsd:attribute name="shininess" type="xsd:float" use="optional"/>
        <xsd:attribute name="specularity" type="xsd:string" use="optional"/>
        <xsd:attribute name="diffusity" type="xsd:string" use="optional"/>
        <xsd:attribute name="metal" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="edge" type="xsd:string" use="optional"/>
        <xsd:attribute name="facet" type="xsd:string" use="optional"/>
        <xsd:attribute name="lightface" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="brightness" type="xsd:string" use="optional"/>
        <xsd:attribute name="lightposition" type="xsd:string" use="optional"/>
        <xsd:attribute name="lightlevel" type="xsd:string" use="optional"/>
        <xsd:attribute name="lightharsh" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="lightposition2" type="xsd:string" use="optional"/>
        <xsd:attribute name="lightlevel2" type="xsd:string" use="optional"/>
        <xsd:attribute name="lightharsh2" type="s:ST_TrueFalse" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Callout">
        <xsd:attributeGroup ref="v:AG_Ext"/>
        <xsd:attribute name="on" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="type" type="xsd:string" use="optional"/>
        <xsd:attribute name="gap" type="xsd:string" use="optional"/>
        <xsd:attribute name="angle" type="ST_Angle" use="optional"/>
        <xsd:attribute name="dropauto" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="drop" type="ST_CalloutDrop" use="optional"/>
        <xsd:attribute name="distance" type="xsd:string" use="optional"/>
        <xsd:attribute name="lengthspecified" type="s:ST_TrueFalse" default="f" use="optional"/>
        <xsd:attribute name="length" type="xsd:string" use="optional"/>
        <xsd:attribute name="accentbar" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="textborder" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="minusx" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="minusy" type="s:ST_TrueFalse" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Lock">
        <xsd:attributeGroup ref="v:AG_Ext"/>
        <xsd:attribute name="position" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="selection" type="s:ST_TrueFalse" use="optional"/>
```



```
        <xsd:attribute name="grouping" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="ungrouping" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="rotation" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="cropping" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="verticies" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="adjusthandles" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="text" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="aspectratio" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="shapetype" type="s:ST_TrueFalse" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OLEObject">
        <xsd:sequence>
            <xsd:element name="LinkType" type="ST_OLELinkType" minOccurs="0"/>
            <xsd:element name="LockedField" type="s:ST_TrueFalseBlank" minOccurs="0"/>
            <xsd:element name="FieldCodes" type="xsd:string" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="Type" type="ST_OLEType" use="optional"/>
        <xsd:attribute name="ProgID" type="xsd:string" use="optional"/>
        <xsd:attribute name="ShapeID" type="xsd:string" use="optional"/>
        <xsd:attribute name="DrawAspect" type="ST_OLEDrawAspect" use="optional"/>
        <xsd:attribute name="ObjectID" type="xsd:string" use="optional"/>
        <xsd:attribute ref="r:id" use="optional"/>
        <xsd:attribute name="UpdateMode" type="ST_OLEUpdateMode" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Complex">
        <xsd:attributeGroup ref="v:AG_Ext"/>
    </xsd:complexType>
    <xsd:complexType name="CT_StrokeChild">
        <xsd:attributeGroup ref="v:AG_Ext"/>
        <xsd:attribute name="on" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="weight" type="xsd:string" use="optional"/>
        <xsd:attribute name="color" type="s:ST_ColorType" use="optional"/>
        <xsd:attribute name="color2" type="s:ST_ColorType" use="optional"/>
        <xsd:attribute name="opacity" type="xsd:string" use="optional"/>
        <xsd:attribute name="linestyle" type="v:ST_StrokeLineStyle" use="optional"/>
        <xsd:attribute name="miterlimit" type="xsd:decimal" use="optional"/>
        <xsd:attribute name="joinstyle" type="v:ST_StrokeJoinStyle" use="optional"/>
        <xsd:attribute name="endcap" type="v:ST_StrokeEndCap" use="optional"/>
        <xsd:attribute name="dashstyle" type="xsd:string" use="optional"/>
        <xsd:attribute name="insetpen" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="filltype" type="v:ST_FillType" use="optional"/>
        <xsd:attribute name="src" type="xsd:string" use="optional"/>
        <xsd:attribute name="imageaspect" type="v:ST_ImageAspect" use="optional"/>
        <xsd:attribute name="imagesize" type="xsd:string" use="optional"/>
        <xsd:attribute name="imagealignshape" type="s:ST_TrueFalse" use="optional"/>
        <xsd:attribute name="startarrow" type="v:ST_StrokeArrowType" use="optional"/>
        <xsd:attribute name="startarrowwidth" type="v:ST_StrokeArrowWidth" use="optional"/>
        <xsd:attribute name="startarrowlength" type="v:ST_StrokeArrowLength" use="optional"/>
        <xsd:attribute name="endarrow" type="v:ST_StrokeArrowType" use="optional"/>
        <xsd:attribute name="endarrowwidth" type="v:ST_StrokeArrowWidth" use="optional"/>
        <xsd:attribute name="endarrowlength" type="v:ST_StrokeArrowLength" use="optional"/>
        <xsd:attribute ref="href"/>
        <xsd:attribute ref="althref"/>
```

```
        <xsd:attribute ref="title"/>
        <xsd:attribute ref="forcedash"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ClipPath">
        <xsd:attribute name="v" type="xsd:string" use="required" form="qualified"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Fill">
        <xsd:attributeGroup ref="v:AG_Ext"/>
        <xsd:attribute name="type" type="ST_FillType"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_RType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="arc"/>
            <xsd:enumeration value="callout"/>
            <xsd:enumeration value="connector"/>
            <xsd:enumeration value="align"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_How">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="top"/>
            <xsd:enumeration value="middle"/>
            <xsd:enumeration value="bottom"/>
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="right"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_BWMode">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="color"/>
            <xsd:enumeration value="auto"/>
            <xsd:enumeration value="grayScale"/>
            <xsd:enumeration value="lightGrayscale"/>
            <xsd:enumeration value="inverseGray"/>
            <xsd:enumeration value="grayOutline"/>
            <xsd:enumeration value="highContrast"/>
            <xsd:enumeration value="black"/>
            <xsd:enumeration value="white"/>
            <xsd:enumeration value="hide"/>
            <xsd:enumeration value="undrawn"/>
            <xsd:enumeration value="blackTextAndLines"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ScreenSize">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="544,376"/>
            <xsd:enumeration value="640,480"/>
            <xsd:enumeration value="720,512"/>
            <xsd:enumeration value="800,600"/>
            <xsd:enumeration value="1024,768"/>
            <xsd:enumeration value="1152,862"/>
        </xsd:restriction>
```



```
    </xsd:simpleType>
    <xsd:simpleType name="ST_InsetMode">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="auto"/>
            <xsd:enumeration value="custom"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ColorMode">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="auto"/>
            <xsd:enumeration value="custom"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ContentType">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_DiagramLayout">
        <xsd:restriction base="xsd:integer">
            <xsd:enumeration value="0"/>
            <xsd:enumeration value="1"/>
            <xsd:enumeration value="2"/>
            <xsd:enumeration value="3"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ExtrusionType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="perspective"/>
            <xsd:enumeration value="parallel"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ExtrusionRender">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="solid"/>
            <xsd:enumeration value="wireFrame"/>
            <xsd:enumeration value="boundingCube"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ExtrusionPlane">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="XY"/>
            <xsd:enumeration value="ZX"/>
            <xsd:enumeration value="YZ"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Angle">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="any"/>
            <xsd:enumeration value="30"/>
            <xsd:enumeration value="45"/>
            <xsd:enumeration value="60"/>
            <xsd:enumeration value="90"/>
            <xsd:enumeration value="auto"/>
        </xsd:restriction>
```

```
    </xsd:simpleType>
    <xsd:simpleType name="ST_CalloutDrop">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_CalloutPlacement">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="top"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="bottom"/>
            <xsd:enumeration value="user"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ConnectorType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="straight"/>
            <xsd:enumeration value="elbow"/>
            <xsd:enumeration value="curved"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_HrAlign">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="right"/>
            <xsd:enumeration value="center"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ConnectType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="rect"/>
            <xsd:enumeration value="segments"/>
            <xsd:enumeration value="custom"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_OLELinkType">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_OLEType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Embed"/>
            <xsd:enumeration value="Link"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_OLEDrawAspect">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Content"/>
            <xsd:enumeration value="Icon"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_OLEUpdateMode">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Always"/>
```



```
            <xsd:enumeration value="OnCall"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FillType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="gradientCenter"/>
            <xsd:enumeration value="solid"/>
            <xsd:enumeration value="pattern"/>
            <xsd:enumeration value="tile"/>
            <xsd:enumeration value="frame"/>
            <xsd:enumeration value="gradientUnscaled"/>
            <xsd:enumeration value="gradientRadial"/>
            <xsd:enumeration value="gradient"/>
            <xsd:enumeration value="background"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>
```

A.6.3 VML - WordprocessingML Drawing

This schema is available in the file vml-wordprocessingDrawing.xsd.

```
<xsd:schema xmlns="urn:schemas-microsoft-com:office:word"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:schemas-microsoft-com:office:word"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xsd:element name="bordertop" type="CT_Border"/>
    <xsd:element name="borderleft" type="CT_Border"/>
    <xsd:element name="borderright" type="CT_Border"/>
    <xsd:element name="borderbottom" type="CT_Border"/>
    <xsd:complexType name="CT_Border">
        <xsd:attribute name="type" type="ST_BorderType" use="optional"/>
        <xsd:attribute name="width" type="xsd:positiveInteger" use="optional"/>
        <xsd:attribute name="shadow" type="ST_BorderShadow" use="optional"/>
    </xsd:complexType>
    <xsd:element name="wrap" type="CT_Wrap"/>
    <xsd:complexType name="CT_Wrap">
        <xsd:attribute name="type" type="ST_WrapType" use="optional"/>
        <xsd:attribute name="side" type="ST_WrapSide" use="optional"/>
        <xsd:attribute name="anchorx" type="ST_HorizontalAnchor" use="optional"/>
        <xsd:attribute name="anchory" type="ST_VerticalAnchor" use="optional"/>
    </xsd:complexType>
    <xsd:element name="anchorlock" type="CT_AnchorLock"/>
    <xsd:complexType name="CT_AnchorLock"/>
    <xsd:simpleType name="ST_BorderType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="single"/>
            <xsd:enumeration value="thick"/>
            <xsd:enumeration value="double"/>
            <xsd:enumeration value="hairline"/>
            <xsd:enumeration value="dot"/>
            <xsd:enumeration value="dash"/>
            <xsd:enumeration value="dotDash"/>
```

```
            <xsd:enumeration value="dashDotDot"/>
            <xsd:enumeration value="triple"/>
            <xsd:enumeration value="thinThickSmall"/>
            <xsd:enumeration value="thickThinSmall"/>
            <xsd:enumeration value="thickBetweenThinSmall"/>
            <xsd:enumeration value="thinThick"/>
            <xsd:enumeration value="thickThin"/>
            <xsd:enumeration value="thickBetweenThin"/>
            <xsd:enumeration value="thinThickLarge"/>
            <xsd:enumeration value="thickThinLarge"/>
            <xsd:enumeration value="thickBetweenThinLarge"/>
            <xsd:enumeration value="wave"/>
            <xsd:enumeration value="doubleWave"/>
            <xsd:enumeration value="dashedSmall"/>
            <xsd:enumeration value="dashDotStroked"/>
            <xsd:enumeration value="threeDEmboss"/>
            <xsd:enumeration value="threeDEngrave"/>
            <xsd:enumeration value="HTMLOutset"/>
            <xsd:enumeration value="HTMLInset"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_BorderShadow">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="t"/>
            <xsd:enumeration value="true"/>
            <xsd:enumeration value="f"/>
            <xsd:enumeration value="false"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_WrapType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="topAndBottom"/>
            <xsd:enumeration value="square"/>
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="tight"/>
            <xsd:enumeration value="through"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_WrapSide">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="both"/>
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="right"/>
            <xsd:enumeration value="largest"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_HorizontalAnchor">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="margin"/>
            <xsd:enumeration value="page"/>
            <xsd:enumeration value="text"/>
            <xsd:enumeration value="char"/>
        </xsd:restriction>
```



```
    </xsd:simpleType>
    <xsd:simpleType name="ST_VerticalAnchor">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="margin"/>
            <xsd:enumeration value="page"/>
            <xsd:enumeration value="text"/>
            <xsd:enumeration value="line"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>
```

A.6.4 VML - SpreadsheetML Drawing

This schema is available in the file vml-spreadsheetDrawing.xsd.

```
<xsd:schema xmlns="urn:schemas-microsoft-com:office:excel"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:s="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
  targetNamespace="urn:schemas-microsoft-com:office:excel" elementFormDefault="qualified"
  attributeFormDefault="unqualified">
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
    <xsd:element name="ClientData" type="CT_ClientData"/>
    <xsd:complexType name="CT_ClientData">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="MoveWithCells" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="SizeWithCells" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="Anchor" type="xsd:string"/>
            <xsd:element name="Locked" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="DefaultSize" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="PrintObject" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="Disabled" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="AutoFill" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="AutoLine" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="AutoPict" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="FmlaMacro" type="xsd:string"/>
            <xsd:element name="TextHAlign" type="xsd:string"/>
            <xsd:element name="TextVAlign" type="xsd:string"/>
            <xsd:element name="LockText" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="JustLastX" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="SecretEdit" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="Default" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="Help" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="Cancel" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="Dismiss" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="Accel" type="xsd:integer"/>
            <xsd:element name="Accel2" type="xsd:integer"/>
            <xsd:element name="Row" type="xsd:integer"/>
            <xsd:element name="Column" type="xsd:integer"/>
            <xsd:element name="Visible" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="RowHidden" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="ColHidden" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="VTEdit" type="xsd:integer"/>
```

```
            <xsd:element name="MultiLine" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="VScroll" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="ValidIds" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="FmlaRange" type="xsd:string"/>
            <xsd:element name="WidthMin" type="xsd:integer"/>
            <xsd:element name="Sel" type="xsd:integer"/>
            <xsd:element name="NoThreeD2" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="SelType" type="xsd:string"/>
            <xsd:element name="MultiSel" type="xsd:string"/>
            <xsd:element name="LCT" type="xsd:string"/>
            <xsd:element name="ListItem" type="xsd:string"/>
            <xsd:element name="DropStyle" type="xsd:string"/>
            <xsd:element name="Colored" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="DropLines" type="xsd:integer"/>
            <xsd:element name="Checked" type="xsd:integer"/>
            <xsd:element name="FmlaLink" type="xsd:string"/>
            <xsd:element name="FmlaPict" type="xsd:string"/>
            <xsd:element name="NoThreeD" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="FirstButton" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="FmlaGroup" type="xsd:string"/>
            <xsd:element name="Val" type="xsd:integer"/>
            <xsd:element name="Min" type="xsd:integer"/>
            <xsd:element name="Max" type="xsd:integer"/>
            <xsd:element name="Inc" type="xsd:integer"/>
            <xsd:element name="Page" type="xsd:integer"/>
            <xsd:element name="Horiz" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="Dx" type="xsd:integer"/>
            <xsd:element name="MapOCX" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="CF" type="ST_CF"/>
            <xsd:element name="Camera" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="RecalcAlways" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="AutoScale" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="DDE" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="UIObj" type="s:ST_TrueFalseBlank"/>
            <xsd:element name="ScriptText" type="xsd:string"/>
            <xsd:element name="ScriptExtended" type="xsd:string"/>
            <xsd:element name="ScriptLanguage" type="xsd:nonNegativeInteger"/>
            <xsd:element name="ScriptLocation" type="xsd:nonNegativeInteger"/>
            <xsd:element name="FmlaTxbx" type="xsd:string"/>
        </xsd:choice>
        <xsd:attribute name="ObjectType" type="ST_ObjectType" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_CF">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ObjectType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Button"/>
            <xsd:enumeration value="Checkbox"/>
            <xsd:enumeration value="Dialog"/>
            <xsd:enumeration value="Drop"/>
            <xsd:enumeration value="Edit"/>
            <xsd:enumeration value="GBox"/>
```



```
            <xsd:enumeration value="Label"/>
            <xsd:enumeration value="LineA"/>
            <xsd:enumeration value="List"/>
            <xsd:enumeration value="Movie"/>
            <xsd:enumeration value="Note"/>
            <xsd:enumeration value="Pict"/>
            <xsd:enumeration value="Radio"/>
            <xsd:enumeration value="RectA"/>
            <xsd:enumeration value="Scroll"/>
            <xsd:enumeration value="Spin"/>
            <xsd:enumeration value="Shape"/>
            <xsd:enumeration value="Group"/>
            <xsd:enumeration value="Rect"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>
```

A.6.5 VML - PresentationML Drawing

This schema is available in the file vml-presentationDrawing.xsd.

```
<xsd:schema xmlns="urn:schemas-microsoft-com:office:powerpoint"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:schemas-microsoft-
  com:office:powerpoint" elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xsd:element name="iscomment" type="CT_Empty"/>
    <xsd:element name="textdata" type="CT_Rel"/>
    <xsd:complexType name="CT_Empty"/>
    <xsd:complexType name="CT_Rel">
        <xsd:attribute name="id" type="xsd:string"/>
    </xsd:complexType>
</xsd:schema>
```

A.7 Shared MLs

A.7.1 Math

This schema is available in the file shared-math.xsd.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.openxmlformats.org/officeDocument/2006/math"
  xmlns:m="http://schemas.openxmlformats.org/officeDocument/2006/math"
  xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main"
  xmlns:s="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
  elementFormDefault="qualified" attributeFormDefault="qualified" blockDefault="#all"
  targetNamespace="http://schemas.openxmlformats.org/officeDocument/2006/math">
    <xsd:import namespace="http://schemas.openxmlformats.org/wordprocessingml/2006/main"
      schemaLocation="wml.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
    <xsd:import namespace="http://www.w3.org/XML/1998/namespace"/>
    <xsd:simpleType name="ST_Integer255">
        <xsd:restriction base="xsd:integer">
            <xsd:minInclusive value="1"/>
```



```
            <xsd:maxInclusive value="255"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Integer255">
        <xsd:attribute name="val" type="ST_Integer255" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Integer2">
        <xsd:restriction base="xsd:integer">
            <xsd:minInclusive value="-2"/>
            <xsd:maxInclusive value="2"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Integer2">
        <xsd:attribute name="val" type="ST_Integer2" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SpacingRule">
        <xsd:restriction base="xsd:integer">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="4"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SpacingRule">
        <xsd:attribute name="val" type="ST_SpacingRule" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_UnSignedInteger">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_UnSignedInteger">
        <xsd:attribute name="val" type="ST_UnSignedInteger" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Char">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="1"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Char">
        <xsd:attribute name="val" type="ST_Char" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OnOff">
        <xsd:attribute name="val" type="s:ST_OnOff"/>
    </xsd:complexType>
    <xsd:complexType name="CT_String">
        <xsd:attribute name="val" type="s:ST_String"/>
    </xsd:complexType>
    <xsd:complexType name="CT_XAlign">
        <xsd:attribute name="val" type="s:ST_XAlign" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_YAlign">
        <xsd:attribute name="val" type="s:ST_YAlign" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Shp">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="centered"/>
```



```
            <xsd:enumeration value="match"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Shp">
        <xsd:attribute name="val" type="ST_Shp" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_FType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="bar"/>
            <xsd:enumeration value="skw"/>
            <xsd:enumeration value="lin"/>
            <xsd:enumeration value="noBar"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_FType">
        <xsd:attribute name="val" type="ST_FType" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_LimLoc">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="undOvr"/>
            <xsd:enumeration value="subSup"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_LimLoc">
        <xsd:attribute name="val" type="ST_LimLoc" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TopBot">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="top"/>
            <xsd:enumeration value="bot"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TopBot">
        <xsd:attribute name="val" type="ST_TopBot" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Script">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="roman"/>
            <xsd:enumeration value="script"/>
            <xsd:enumeration value="fraktur"/>
            <xsd:enumeration value="double-struck"/>
            <xsd:enumeration value="sans-serif"/>
            <xsd:enumeration value="monospace"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Script">
        <xsd:attribute name="val" type="ST_Script"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Style">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="p"/>
            <xsd:enumeration value="b"/>
            <xsd:enumeration value="i"/>
```

```
            <xsd:enumeration value="bi"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Style">
        <xsd:attribute name="val" type="ST_Style"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ManualBreak">
        <xsd:attribute name="alnAt" type="ST_Integer255"/>
    </xsd:complexType>
    <xsd:group name="EG_ScriptStyle">
        <xsd:sequence>
            <xsd:element name="scr" minOccurs="0" type="CT_Script"/>
            <xsd:element name="sty" minOccurs="0" type="CT_Style"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_RPR">
        <xsd:sequence>
            <xsd:element name="lit" minOccurs="0" type="CT_OnOff"/>
            <xsd:choice>
                <xsd:element name="nor" minOccurs="0" type="CT_OnOff"/>
                <xsd:sequence>
                    <xsd:group ref="EG_ScriptStyle"/>
                </xsd:sequence>
            </xsd:choice>
            <xsd:element name="brk" minOccurs="0" type="CT_ManualBreak"/>
            <xsd:element name="aln" minOccurs="0" type="CT_OnOff"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Text">
        <xsd:simpleContent>
            <xsd:extension base="s:ST_String">
                <xsd:attribute ref="xml:space" use="optional"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:complexType name="CT_R">
        <xsd:sequence>
            <xsd:element name="rPr" type="CT_RPR" minOccurs="0"/>
            <xsd:group ref="w:EG_RPr" minOccurs="0"/>
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:group ref="w:EG_RunInnerContent"/>
                <xsd:element name="t" type="CT_Text" minOccurs="0"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CtrlPr">
        <xsd:sequence>
            <xsd:group ref="w:EG_RPrMath" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_AccPr">
        <xsd:sequence>
            <xsd:element name="chr" type="CT_Char" minOccurs="0"/>
```



```
            <xsd:element name="ctrlPr" type="CT_CtrlPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Acc">
        <xsd:sequence>
            <xsd:element name="accPr" type="CT_AccPr" minOccurs="0"/>
            <xsd:element name="e" type="CT_OMathArg"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_BarPr">
        <xsd:sequence>
            <xsd:element name="pos" type="CT_TopBot" minOccurs="0"/>
            <xsd:element name="ctrlPr" type="CT_CtrlPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Bar">
        <xsd:sequence>
            <xsd:element name="barPr" type="CT_BarPr" minOccurs="0"/>
            <xsd:element name="e" type="CT_OMathArg"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_BoxPr">
        <xsd:sequence>
            <xsd:element name="opEmu" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="noBreak" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="diff" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="brk" type="CT_ManualBreak" minOccurs="0"/>
            <xsd:element name="aln" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="ctrlPr" type="CT_CtrlPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Box">
        <xsd:sequence>
            <xsd:element name="boxPr" type="CT_BoxPr" minOccurs="0"/>
            <xsd:element name="e" type="CT_OMathArg"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_BorderBoxPr">
        <xsd:sequence>
            <xsd:element name="hideTop" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="hideBot" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="hideLeft" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="hideRight" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="strikeH" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="strikeV" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="strikeBLTR" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="strikeTLBR" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="ctrlPr" type="CT_CtrlPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_BorderBox">
        <xsd:sequence>
            <xsd:element name="borderBoxPr" type="CT_BorderBoxPr" minOccurs="0"/>
```

```
            <xsd:element name="e" type="CT_OMathArg"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DPr">
        <xsd:sequence>
            <xsd:element name="begChr" type="CT_Char" minOccurs="0"/>
            <xsd:element name="sepChr" type="CT_Char" minOccurs="0"/>
            <xsd:element name="endChr" type="CT_Char" minOccurs="0"/>
            <xsd:element name="grow" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="shp" type="CT_Shp" minOccurs="0"/>
            <xsd:element name="ctrlPr" type="CT_CtrlPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_D">
        <xsd:sequence>
            <xsd:element name="dPr" type="CT_DPr" minOccurs="0"/>
            <xsd:element name="e" type="CT_OMathArg" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_EqArrPr">
        <xsd:sequence>
            <xsd:element name="baseJc" type="CT_YAlign" minOccurs="0"/>
            <xsd:element name="maxDist" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="objDist" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="rSpRule" type="CT_SpacingRule" minOccurs="0"/>
            <xsd:element name="rSp" type="CT_UnSignedInteger" minOccurs="0"/>
            <xsd:element name="ctrlPr" type="CT_CtrlPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_EqArr">
        <xsd:sequence>
            <xsd:element name="eqArrPr" type="CT_EqArrPr" minOccurs="0"/>
            <xsd:element name="e" type="CT_OMathArg" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_FPr">
        <xsd:sequence>
            <xsd:element name="type" type="CT_FType" minOccurs="0"/>
            <xsd:element name="ctrlPr" type="CT_CtrlPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_F">
        <xsd:sequence>
            <xsd:element name="fPr" type="CT_FPr" minOccurs="0"/>
            <xsd:element name="num" type="CT_OMathArg"/>
            <xsd:element name="den" type="CT_OMathArg"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_FuncPr">
        <xsd:sequence>
            <xsd:element name="ctrlPr" type="CT_CtrlPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
```



```
    <xsd:complexType name="CT_Func">
        <xsd:sequence>
            <xsd:element name="funcPr" type="CT_FuncPr" minOccurs="0"/>
            <xsd:element name="fName" type="CT_OMathArg"/>
            <xsd:element name="e" type="CT_OMathArg"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupChrPr">
        <xsd:sequence>
            <xsd:element name="chr" type="CT_Char" minOccurs="0"/>
            <xsd:element name="pos" type="CT_TopBot" minOccurs="0"/>
            <xsd:element name="vertJc" type="CT_TopBot" minOccurs="0"/>
            <xsd:element name="ctrlPr" type="CT_CtrlPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupChr">
        <xsd:sequence>
            <xsd:element name="groupChrPr" type="CT_GroupChrPr" minOccurs="0"/>
            <xsd:element name="e" type="CT_OMathArg"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_LimLowPr">
        <xsd:sequence>
            <xsd:element name="ctrlPr" type="CT_CtrlPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_LimLow">
        <xsd:sequence>
            <xsd:element name="limLowPr" type="CT_LimLowPr" minOccurs="0"/>
            <xsd:element name="e" type="CT_OMathArg"/>
            <xsd:element name="lim" type="CT_OMathArg"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_LimUppPr">
        <xsd:sequence>
            <xsd:element name="ctrlPr" type="CT_CtrlPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_LimUpp">
        <xsd:sequence>
            <xsd:element name="limUppPr" type="CT_LimUppPr" minOccurs="0"/>
            <xsd:element name="e" type="CT_OMathArg"/>
            <xsd:element name="lim" type="CT_OMathArg"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_MCPr">
        <xsd:sequence>
            <xsd:element name="count" type="CT_Integer255" minOccurs="0"/>
            <xsd:element name="mcJc" type="CT_XAlign" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_MC">
        <xsd:sequence>
```

```
            <xsd:element name="mcPr" type="CT_MCPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_MCS">
        <xsd:sequence>
            <xsd:element name="mc" type="CT_MC" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_MPr">
        <xsd:sequence>
            <xsd:element name="baseJc" type="CT_YAlign" minOccurs="0"/>
            <xsd:element name="plcHide" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="rSpRule" type="CT_SpacingRule" minOccurs="0"/>
            <xsd:element name="cGpRule" type="CT_SpacingRule" minOccurs="0"/>
            <xsd:element name="rSp" type="CT_UnSignedInteger" minOccurs="0"/>
            <xsd:element name="cSp" type="CT_UnSignedInteger" minOccurs="0"/>
            <xsd:element name="cGp" type="CT_UnSignedInteger" minOccurs="0"/>
            <xsd:element name="mcs" type="CT_MCS" minOccurs="0"/>
            <xsd:element name="ctrlPr" type="CT_CtrlPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_MR">
        <xsd:sequence>
            <xsd:element name="e" type="CT_OMathArg" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_M">
        <xsd:sequence>
            <xsd:element name="mPr" type="CT_MPr" minOccurs="0"/>
            <xsd:element name="mr" type="CT_MR" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_NaryPr">
        <xsd:sequence>
            <xsd:element name="chr" type="CT_Char" minOccurs="0"/>
            <xsd:element name="limLoc" type="CT_LimLoc" minOccurs="0"/>
            <xsd:element name="grow" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="subHide" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="supHide" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="ctrlPr" type="CT_CtrlPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Nary">
        <xsd:sequence>
            <xsd:element name="naryPr" type="CT_NaryPr" minOccurs="0"/>
            <xsd:element name="sub" type="CT_OMathArg"/>
            <xsd:element name="sup" type="CT_OMathArg"/>
            <xsd:element name="e" type="CT_OMathArg"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_PhantPr">
        <xsd:sequence>
            <xsd:element name="show" type="CT_OnOff" minOccurs="0"/>
```



```
            <xsd:element name="zeroWid" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="zeroAsc" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="zeroDesc" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="transp" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="ctrlPr" type="CT_CtrlPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Phant">
        <xsd:sequence>
            <xsd:element name="phantPr" type="CT_PhantPr" minOccurs="0"/>
            <xsd:element name="e" type="CT_OMathArg"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_RadPr">
        <xsd:sequence>
            <xsd:element name="degHide" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="ctrlPr" type="CT_CtrlPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Rad">
        <xsd:sequence>
            <xsd:element name="radPr" type="CT_RadPr" minOccurs="0"/>
            <xsd:element name="deg" type="CT_OMathArg"/>
            <xsd:element name="e" type="CT_OMathArg"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SPrePr">
        <xsd:sequence>
            <xsd:element name="ctrlPr" type="CT_CtrlPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SPre">
        <xsd:sequence>
            <xsd:element name="sPrePr" type="CT_SPrePr" minOccurs="0"/>
            <xsd:element name="sub" type="CT_OMathArg"/>
            <xsd:element name="sup" type="CT_OMathArg"/>
            <xsd:element name="e" type="CT_OMathArg"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SSubPr">
        <xsd:sequence>
            <xsd:element name="ctrlPr" type="CT_CtrlPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SSub">
        <xsd:sequence>
            <xsd:element name="sSubPr" type="CT_SSubPr" minOccurs="0"/>
            <xsd:element name="e" type="CT_OMathArg"/>
            <xsd:element name="sub" type="CT_OMathArg"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SSubSupPr">
        <xsd:sequence>
```

```
            <xsd:element name="alnScr" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="ctrlPr" type="CT_CtrlPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SSubSup">
        <xsd:sequence>
            <xsd:element name="sSubSupPr" type="CT_SSubSupPr" minOccurs="0"/>
            <xsd:element name="e" type="CT_OMathArg"/>
            <xsd:element name="sub" type="CT_OMathArg"/>
            <xsd:element name="sup" type="CT_OMathArg"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SSupPr">
        <xsd:sequence>
            <xsd:element name="ctrlPr" type="CT_CtrlPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SSup">
        <xsd:sequence>
            <xsd:element name="sSupPr" type="CT_SSupPr" minOccurs="0"/>
            <xsd:element name="e" type="CT_OMathArg"/>
            <xsd:element name="sup" type="CT_OMathArg"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_OMathMathElements">
        <xsd:choice>
            <xsd:element name="acc" type="CT_Acc"/>
            <xsd:element name="bar" type="CT_Bar"/>
            <xsd:element name="box" type="CT_Box"/>
            <xsd:element name="borderBox" type="CT_BorderBox"/>
            <xsd:element name="d" type="CT_D"/>
            <xsd:element name="eqArr" type="CT_EqArr"/>
            <xsd:element name="f" type="CT_F"/>
            <xsd:element name="func" type="CT_Func"/>
            <xsd:element name="groupChr" type="CT_GroupChr"/>
            <xsd:element name="limLow" type="CT_LimLow"/>
            <xsd:element name="limUpp" type="CT_LimUpp"/>
            <xsd:element name="m" type="CT_M"/>
            <xsd:element name="nary" type="CT_Nary"/>
            <xsd:element name="phant" type="CT_Phant"/>
            <xsd:element name="rad" type="CT_Rad"/>
            <xsd:element name="sPre" type="CT_SPre"/>
            <xsd:element name="sSub" type="CT_SSub"/>
            <xsd:element name="sSubSup" type="CT_SSubSup"/>
            <xsd:element name="sSup" type="CT_SSup"/>
            <xsd:element name="r" type="CT_R"/>
        </xsd:choice>
    </xsd:group>
    <xsd:group name="EG_OMathElements">
        <xsd:choice>
            <xsd:group ref="EG_OMathMathElements"/>
            <xsd:group ref="w:EG_PContentMath"/>
        </xsd:choice>
```



```
    </xsd:group>
    <xsd:complexType name="CT_OMathArgPr">
        <xsd:sequence>
            <xsd:element name="argSz" type="CT_Integer2" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_OMathArg">
        <xsd:sequence>
            <xsd:element name="argPr" type="CT_OMathArgPr" minOccurs="0"/>
            <xsd:group ref="EG_OMathElements" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="ctrlPr" type="CT_CtrlPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_Jc">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="right"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="centerGroup"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_OMathJc">
        <xsd:attribute name="val" type="ST_Jc"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OMathParaPr">
        <xsd:sequence>
            <xsd:element name="jc" type="CT_OMathJc" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TwipsMeasure">
        <xsd:attribute name="val" type="s:ST_TwipsMeasure" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_BreakBin">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="before"/>
            <xsd:enumeration value="after"/>
            <xsd:enumeration value="repeat"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_BreakBin">
        <xsd:attribute name="val" type="ST_BreakBin"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_BreakBinSub">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="--"/>
            <xsd:enumeration value="-+"/>
            <xsd:enumeration value="+-"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_BreakBinSub">
        <xsd:attribute name="val" type="ST_BreakBinSub"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MathPr">
```

```
        <xsd:sequence>
            <xsd:element name="mathFont" type="CT_String" minOccurs="0"/>
            <xsd:element name="brkBin" type="CT_BreakBin" minOccurs="0"/>
            <xsd:element name="brkBinSub" type="CT_BreakBinSub" minOccurs="0"/>
            <xsd:element name="smallFrac" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="dispDef" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="lMargin" type="CT_TwipsMeasure" minOccurs="0"/>
            <xsd:element name="rMargin" type="CT_TwipsMeasure" minOccurs="0"/>
            <xsd:element name="defJc" type="CT_OMathJc" minOccurs="0"/>
            <xsd:element name="preSp" type="CT_TwipsMeasure" minOccurs="0"/>
            <xsd:element name="postSp" type="CT_TwipsMeasure" minOccurs="0"/>
            <xsd:element name="interSp" type="CT_TwipsMeasure" minOccurs="0"/>
            <xsd:element name="intraSp" type="CT_TwipsMeasure" minOccurs="0"/>
            <xsd:choice minOccurs="0">
                <xsd:element name="wrapIndent" type="CT_TwipsMeasure"/>
                <xsd:element name="wrapRight" type="CT_OnOff"/>
            </xsd:choice>
            <xsd:element name="intLim" type="CT_LimLoc" minOccurs="0"/>
            <xsd:element name="naryLim" type="CT_LimLoc" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="mathPr" type="CT_MathPr"/>
    <xsd:complexType name="CT_OMathPara">
        <xsd:sequence>
            <xsd:element name="oMathParaPr" type="CT_OMathParaPr" minOccurs="0"/>
            <xsd:element name="oMath" type="CT_OMath" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_OMath">
        <xsd:sequence>
            <xsd:group ref="EG_OMathElements" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="oMathPara" type="CT_OMathPara"/>
    <xsd:element name="oMath" type="CT_OMath"/>
</xsd:schema>
```

A.7.2 Extended Properties

This schema is available in the file shared-documentPropertiesExtended.xsd.

```
<xsd:schema xmlns="http://schemas.openxmlformats.org/officeDocument/2006/extended-properties"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:vt="http://schemas.openxmlformats.org/officeDocument/2006/docPropsVTypes"
  targetNamespace="http://schemas.openxmlformats.org/officeDocument/2006/extended-properties"
  elementFormDefault="qualified" blockDefault="#all">
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/docPropsVTypes"
      schemaLocation="shared-documentPropertiesVariantTypes.xsd"/>
    <xsd:element name="Properties" type="CT_Properties"/>
    <xsd:complexType name="CT_Properties">
        <xsd:all>
            <xsd:element name="Template" minOccurs="0" maxOccurs="1" type="xsd:string"/>
            <xsd:element name="Manager" minOccurs="0" maxOccurs="1" type="xsd:string"/>
```



```
            <xsd:element name="Company" minOccurs="0" maxOccurs="1" type="xsd:string"/>
            <xsd:element name="Pages" minOccurs="0" maxOccurs="1" type="xsd:int"/>
            <xsd:element name="Words" minOccurs="0" maxOccurs="1" type="xsd:int"/>
            <xsd:element name="Characters" minOccurs="0" maxOccurs="1" type="xsd:int"/>
            <xsd:element name="PresentationFormat" minOccurs="0" maxOccurs="1" type="xsd:string"/>
            <xsd:element name="Lines" minOccurs="0" maxOccurs="1" type="xsd:int"/>
            <xsd:element name="Paragraphs" minOccurs="0" maxOccurs="1" type="xsd:int"/>
            <xsd:element name="Slides" minOccurs="0" maxOccurs="1" type="xsd:int"/>
            <xsd:element name="Notes" minOccurs="0" maxOccurs="1" type="xsd:int"/>
            <xsd:element name="TotalTime" minOccurs="0" maxOccurs="1" type="xsd:int"/>
            <xsd:element name="HiddenSlides" minOccurs="0" maxOccurs="1" type="xsd:int"/>
            <xsd:element name="MMClips" minOccurs="0" maxOccurs="1" type="xsd:int"/>
            <xsd:element name="ScaleCrop" minOccurs="0" maxOccurs="1" type="xsd:boolean"/>
            <xsd:element name="HeadingPairs" minOccurs="0" maxOccurs="1" type="CT_VectorVariant"/>
            <xsd:element name="TitlesOfParts" minOccurs="0" maxOccurs="1" type="CT_VectorLpstr"/>
            <xsd:element name="LinksUpToDate" minOccurs="0" maxOccurs="1" type="xsd:boolean"/>
            <xsd:element name="CharactersWithSpaces" minOccurs="0" maxOccurs="1" type="xsd:int"/>
            <xsd:element name="SharedDoc" minOccurs="0" maxOccurs="1" type="xsd:boolean"/>
            <xsd:element name="HyperlinkBase" minOccurs="0" maxOccurs="1" type="xsd:string"/>
            <xsd:element name="HLinks" minOccurs="0" maxOccurs="1" type="CT_VectorVariant"/>
            <xsd:element name="HyperlinksChanged" minOccurs="0" maxOccurs="1" type="xsd:boolean"/>
            <xsd:element name="DigSig" minOccurs="0" maxOccurs="1" type="CT_DigSigBlob"/>
            <xsd:element name="Application" minOccurs="0" maxOccurs="1" type="xsd:string"/>
            <xsd:element name="AppVersion" minOccurs="0" maxOccurs="1" type="xsd:string"/>
            <xsd:element name="DocSecurity" minOccurs="0" maxOccurs="1" type="xsd:int"/>
        </xsd:all>
    </xsd:complexType>
    <xsd:complexType name="CT_VectorVariant">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element ref="vt:vector"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_VectorLpstr">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element ref="vt:vector"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DigSigBlob">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element ref="vt:blob"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

A.7.3 Custom Properties

This schema is available in the file shared-documentPropertiesCustom.xsd.

```
<xsd:schema xmlns="http://schemas.openxmlformats.org/officeDocument/2006/custom-properties"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:vt="http://schemas.openxmlformats.org/officeDocument/2006/docPropsVTypes"
  xmlns:s="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
```



```
  targetNamespace="http://schemas.openxmlformats.org/officeDocument/2006/custom-properties"
  blockDefault="#all" elementFormDefault="qualified">
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/docPropsVTypes"
      schemaLocation="shared-documentPropertiesVariantTypes.xsd"/>
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
    <xsd:element name="Properties" type="CT_Properties"/>
    <xsd:complexType name="CT_Properties">
        <xsd:sequence>
            <xsd:element name="property" minOccurs="0" maxOccurs="unbounded" type="CT_Property"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Property">
        <xsd:choice minOccurs="1" maxOccurs="1">
            <xsd:element ref="vt:vector"/>
            <xsd:element ref="vt:array"/>
            <xsd:element ref="vt:blob"/>
            <xsd:element ref="vt:oblob"/>
            <xsd:element ref="vt:empty"/>
            <xsd:element ref="vt:null"/>
            <xsd:element ref="vt:i1"/>
            <xsd:element ref="vt:i2"/>
            <xsd:element ref="vt:i4"/>
            <xsd:element ref="vt:i8"/>
            <xsd:element ref="vt:int"/>
            <xsd:element ref="vt:ui1"/>
            <xsd:element ref="vt:ui2"/>
            <xsd:element ref="vt:ui4"/>
            <xsd:element ref="vt:ui8"/>
            <xsd:element ref="vt:uint"/>
            <xsd:element ref="vt:r4"/>
            <xsd:element ref="vt:r8"/>
            <xsd:element ref="vt:decimal"/>
            <xsd:element ref="vt:lpstr"/>
            <xsd:element ref="vt:lpwstr"/>
            <xsd:element ref="vt:bstr"/>
            <xsd:element ref="vt:date"/>
            <xsd:element ref="vt:filetime"/>
            <xsd:element ref="vt:bool"/>
            <xsd:element ref="vt:cy"/>
            <xsd:element ref="vt:error"/>
            <xsd:element ref="vt:stream"/>
            <xsd:element ref="vt:ostream"/>
            <xsd:element ref="vt:storage"/>
            <xsd:element ref="vt:ostorage"/>
            <xsd:element ref="vt:vstream"/>
            <xsd:element ref="vt:clsid"/>
        </xsd:choice>
        <xsd:attribute name="fmtid" use="required" type="s:ST_Guid"/>
        <xsd:attribute name="pid" use="required" type="xsd:int"/>
        <xsd:attribute name="name" use="optional" type="xsd:string"/>
        <xsd:attribute name="linkTarget" use="optional" type="xsd:string"/>
    </xsd:complexType>
```



```
</xsd:schema>
```

A.7.4 Variant Types

This schema is available in the file shared-documentPropertiesVariantTypes.xsd.

```
<xsd:schema xmlns="http://schemas.openxmlformats.org/officeDocument/2006/docPropsVTypes"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:s="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
  targetNamespace="http://schemas.openxmlformats.org/officeDocument/2006/docPropsVTypes"
  blockDefault="#all" elementFormDefault="qualified">
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
    <xsd:simpleType name="ST_VectorBaseType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="variant"/>
            <xsd:enumeration value="i1"/>
            <xsd:enumeration value="i2"/>
            <xsd:enumeration value="i4"/>
            <xsd:enumeration value="i8"/>
            <xsd:enumeration value="ui1"/>
            <xsd:enumeration value="ui2"/>
            <xsd:enumeration value="ui4"/>
            <xsd:enumeration value="ui8"/>
            <xsd:enumeration value="r4"/>
            <xsd:enumeration value="r8"/>
            <xsd:enumeration value="lpstr"/>
            <xsd:enumeration value="lpwstr"/>
            <xsd:enumeration value="bstr"/>
            <xsd:enumeration value="date"/>
            <xsd:enumeration value="filetime"/>
            <xsd:enumeration value="bool"/>
            <xsd:enumeration value="cy"/>
            <xsd:enumeration value="error"/>
            <xsd:enumeration value="clsid"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ArrayBaseType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="variant"/>
            <xsd:enumeration value="i1"/>
            <xsd:enumeration value="i2"/>
            <xsd:enumeration value="i4"/>
            <xsd:enumeration value="int"/>
            <xsd:enumeration value="ui1"/>
            <xsd:enumeration value="ui2"/>
            <xsd:enumeration value="ui4"/>
            <xsd:enumeration value="uint"/>
            <xsd:enumeration value="r4"/>
            <xsd:enumeration value="r8"/>
            <xsd:enumeration value="decimal"/>
            <xsd:enumeration value="bstr"/>
            <xsd:enumeration value="date"/>
```

```
            <xsd:enumeration value="bool"/>
            <xsd:enumeration value="cy"/>
            <xsd:enumeration value="error"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Cy">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="\s*[0-9]*\.[0-9]{4}\s*"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Error">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="\s*0x[0-9A-Za-z]{8}\s*"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Empty"/>
    <xsd:complexType name="CT_Null"/>
    <xsd:complexType name="CT_Vector">
        <xsd:choice minOccurs="1" maxOccurs="unbounded">
            <xsd:element ref="variant"/>
            <xsd:element ref="i1"/>
            <xsd:element ref="i2"/>
            <xsd:element ref="i4"/>
            <xsd:element ref="i8"/>
            <xsd:element ref="ui1"/>
            <xsd:element ref="ui2"/>
            <xsd:element ref="ui4"/>
            <xsd:element ref="ui8"/>
            <xsd:element ref="r4"/>
            <xsd:element ref="r8"/>
            <xsd:element ref="lpstr"/>
            <xsd:element ref="lpwstr"/>
            <xsd:element ref="bstr"/>
            <xsd:element ref="date"/>
            <xsd:element ref="filetime"/>
            <xsd:element ref="bool"/>
            <xsd:element ref="cy"/>
            <xsd:element ref="error"/>
            <xsd:element ref="clsid"/>
        </xsd:choice>
        <xsd:attribute name="baseType" type="ST_VectorBaseType" use="required"/>
        <xsd:attribute name="size" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Array">
        <xsd:choice minOccurs="1" maxOccurs="unbounded">
            <xsd:element ref="variant"/>
            <xsd:element ref="i1"/>
            <xsd:element ref="i2"/>
            <xsd:element ref="i4"/>
            <xsd:element ref="int"/>
            <xsd:element ref="ui1"/>
            <xsd:element ref="ui2"/>
            <xsd:element ref="ui4"/>
```



```
            <xsd:element ref="uint"/>
            <xsd:element ref="r4"/>
            <xsd:element ref="r8"/>
            <xsd:element ref="decimal"/>
            <xsd:element ref="bstr"/>
            <xsd:element ref="date"/>
            <xsd:element ref="bool"/>
            <xsd:element ref="error"/>
            <xsd:element ref="cy"/>
        </xsd:choice>
        <xsd:attribute name="lBounds" type="xsd:int" use="required"/>
        <xsd:attribute name="uBounds" type="xsd:int" use="required"/>
        <xsd:attribute name="baseType" type="ST_ArrayBaseType" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Variant">
        <xsd:choice minOccurs="1" maxOccurs="1">
            <xsd:element ref="variant"/>
            <xsd:element ref="vector"/>
            <xsd:element ref="array"/>
            <xsd:element ref="blob"/>
            <xsd:element ref="oblob"/>
            <xsd:element ref="empty"/>
            <xsd:element ref="null"/>
            <xsd:element ref="i1"/>
            <xsd:element ref="i2"/>
            <xsd:element ref="i4"/>
            <xsd:element ref="i8"/>
            <xsd:element ref="int"/>
            <xsd:element ref="ui1"/>
            <xsd:element ref="ui2"/>
            <xsd:element ref="ui4"/>
            <xsd:element ref="ui8"/>
            <xsd:element ref="uint"/>
            <xsd:element ref="r4"/>
            <xsd:element ref="r8"/>
            <xsd:element ref="decimal"/>
            <xsd:element ref="lpstr"/>
            <xsd:element ref="lpwstr"/>
            <xsd:element ref="bstr"/>
            <xsd:element ref="date"/>
            <xsd:element ref="filetime"/>
            <xsd:element ref="bool"/>
            <xsd:element ref="cy"/>
            <xsd:element ref="error"/>
            <xsd:element ref="stream"/>
            <xsd:element ref="ostream"/>
            <xsd:element ref="storage"/>
            <xsd:element ref="ostorage"/>
            <xsd:element ref="vstream"/>
            <xsd:element ref="clsid"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_Vstream">
```

```
        <xsd:simpleContent>
            <xsd:extension base="xsd:base64Binary">
                <xsd:attribute name="version" type="s:ST_Guid"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:element name="variant" type="CT_Variant"/>
    <xsd:element name="vector" type="CT_Vector"/>
    <xsd:element name="array" type="CT_Array"/>
    <xsd:element name="blob" type="xsd:base64Binary"/>
    <xsd:element name="oblob" type="xsd:base64Binary"/>
    <xsd:element name="empty" type="CT_Empty"/>
    <xsd:element name="null" type="CT_Null"/>
    <xsd:element name="i1" type="xsd:byte"/>
    <xsd:element name="i2" type="xsd:short"/>
    <xsd:element name="i4" type="xsd:int"/>
    <xsd:element name="i8" type="xsd:long"/>
    <xsd:element name="int" type="xsd:int"/>
    <xsd:element name="ui1" type="xsd:unsignedByte"/>
    <xsd:element name="ui2" type="xsd:unsignedShort"/>
    <xsd:element name="ui4" type="xsd:unsignedInt"/>
    <xsd:element name="ui8" type="xsd:unsignedLong"/>
    <xsd:element name="uint" type="xsd:unsignedInt"/>
    <xsd:element name="r4" type="xsd:float"/>
    <xsd:element name="r8" type="xsd:double"/>
    <xsd:element name="decimal" type="xsd:decimal"/>
    <xsd:element name="lpstr" type="xsd:string"/>
    <xsd:element name="lpwstr" type="xsd:string"/>
    <xsd:element name="bstr" type="xsd:string"/>
    <xsd:element name="date" type="xsd:dateTime"/>
    <xsd:element name="filetime" type="xsd:dateTime"/>
    <xsd:element name="bool" type="xsd:boolean"/>
    <xsd:element name="cy" type="ST_Cy"/>
    <xsd:element name="error" type="ST_Error"/>
    <xsd:element name="stream" type="xsd:base64Binary"/>
    <xsd:element name="ostream" type="xsd:base64Binary"/>
    <xsd:element name="storage" type="xsd:base64Binary"/>
    <xsd:element name="ostorage" type="xsd:base64Binary"/>
    <xsd:element name="vstream" type="CT_Vstream"/>
    <xsd:element name="clsid" type="s:ST_Guid"/>
</xsd:schema>
```

A.7.5 Custom XML Data Properties

This schema is available in the file shared-customXmlDataProperties.xsd.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.openxmlformats.org/officeDocument/2006/customXml"
  xmlns:s="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
  targetNamespace="http://schemas.openxmlformats.org/officeDocument/2006/customXml"
  elementFormDefault="qualified" attributeFormDefault="qualified" blockDefault="#all">
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
```



```
    <xsd:complexType name="CT_DatastoreSchemaRef">
        <xsd:attribute name="uri" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DatastoreSchemaRefs">
        <xsd:sequence>
            <xsd:element name="schemaRef" type="CT_DatastoreSchemaRef" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DatastoreItem">
        <xsd:sequence>
            <xsd:element name="schemaRefs" type="CT_DatastoreSchemaRefs" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="itemID" type="s:ST_Guid" use="required"/>
    </xsd:complexType>
    <xsd:element name="datastoreItem" type="CT_DatastoreItem"/>
</xsd:schema>
```

A.7.6 Bibliography

This schema is available in the file shared-bibliography.xsd.

```
<xsd:schema xmlns="http://schemas.openxmlformats.org/officeDocument/2006/bibliography"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:s="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
  targetNamespace="http://schemas.openxmlformats.org/officeDocument/2006/bibliography"
  elementFormDefault="qualified">
    <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
    <xsd:simpleType name="ST_SourceType">
        <xsd:restriction base="s:ST_String">
            <xsd:enumeration value="ArticleInAPeriodical"/>
            <xsd:enumeration value="Book"/>
            <xsd:enumeration value="BookSection"/>
            <xsd:enumeration value="JournalArticle"/>
            <xsd:enumeration value="ConferenceProceedings"/>
            <xsd:enumeration value="Report"/>
            <xsd:enumeration value="SoundRecording"/>
            <xsd:enumeration value="Performance"/>
            <xsd:enumeration value="Art"/>
            <xsd:enumeration value="DocumentFromInternetSite"/>
            <xsd:enumeration value="InternetSite"/>
            <xsd:enumeration value="Film"/>
            <xsd:enumeration value="Interview"/>
            <xsd:enumeration value="Patent"/>
            <xsd:enumeration value="ElectronicSource"/>
            <xsd:enumeration value="Case"/>
            <xsd:enumeration value="Misc"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_NameListType">
        <xsd:sequence>
            <xsd:element name="Person" type="CT_PersonType" minOccurs="1" maxOccurs="unbounded"/>
```



```
32          </xsd:sequence>
33      </xsd:complexType>
34      <xsd:complexType name="CT_PersonType">
35          <xsd:sequence>
36              <xsd:element name="Last" type="s:ST_String" minOccurs="0" maxOccurs="unbounded"/>
37              <xsd:element name="First" type="s:ST_String" minOccurs="0" maxOccurs="unbounded"/>
38              <xsd:element name="Middle" type="s:ST_String" minOccurs="0" maxOccurs="unbounded"/>
39          </xsd:sequence>
40      </xsd:complexType>
41      <xsd:complexType name="CT_NameType">
42          <xsd:sequence>
43              <xsd:element name="NameList" type="CT_NameListType" minOccurs="1" maxOccurs="1"/>
44          </xsd:sequence>
45      </xsd:complexType>
46      <xsd:complexType name="CT_NameOrCorporateType">
47          <xsd:sequence>
48              <xsd:choice minOccurs="0" maxOccurs="1">
49                  <xsd:element name="NameList" type="CT_NameListType" minOccurs="1" maxOccurs="1"/>
50                  <xsd:element name="Corporate" minOccurs="1" maxOccurs="1" type="s:ST_String"/>
51              </xsd:choice>
52          </xsd:sequence>
53      </xsd:complexType>
54      <xsd:complexType name="CT_AuthorType">
55          <xsd:sequence>
56              <xsd:choice minOccurs="0" maxOccurs="unbounded">
57                  <xsd:element name="Artist" type="CT_NameType"/>
58                  <xsd:element name="Author" type="CT_NameOrCorporateType"/>
59                  <xsd:element name="BookAuthor" type="CT_NameType"/>
60                  <xsd:element name="Compiler" type="CT_NameType"/>
61                  <xsd:element name="Composer" type="CT_NameType"/>
62                  <xsd:element name="Conductor" type="CT_NameType"/>
63                  <xsd:element name="Counsel" type="CT_NameType"/>
64                  <xsd:element name="Director" type="CT_NameType"/>
65                  <xsd:element name="Editor" type="CT_NameType"/>
66                  <xsd:element name="Interviewee" type="CT_NameType"/>
67                  <xsd:element name="Interviewer" type="CT_NameType"/>
68                  <xsd:element name="Inventor" type="CT_NameType"/>
69                  <xsd:element name="Performer" type="CT_NameOrCorporateType"/>
70                  <xsd:element name="ProducerName" type="CT_NameType"/>
71                  <xsd:element name="Translator" type="CT_NameType"/>
72                  <xsd:element name="Writer" type="CT_NameType"/>
73              </xsd:choice>
74          </xsd:sequence>
75      </xsd:complexType>
76      <xsd:complexType name="CT_SourceType">
77          <xsd:sequence>
78              <xsd:choice minOccurs="0" maxOccurs="unbounded">
79                  <xsd:element name="AbbreviatedCaseNumber" type="s:ST_String"/>
80                  <xsd:element name="AlbumTitle" type="s:ST_String"/>
81                  <xsd:element name="Author" type="CT_AuthorType"/>
82                  <xsd:element name="BookTitle" type="s:ST_String"/>
83                  <xsd:element name="Broadcaster" type="s:ST_String"/>
84                  <xsd:element name="BroadcastTitle" type="s:ST_String"/>
```



```
                <xsd:element name="CaseNumber" type="s:ST_String"/>
                <xsd:element name="ChapterNumber" type="s:ST_String"/>
                <xsd:element name="City" type="s:ST_String"/>
                <xsd:element name="Comments" type="s:ST_String"/>
                <xsd:element name="ConferenceName" type="s:ST_String"/>
                <xsd:element name="CountryRegion" type="s:ST_String"/>
                <xsd:element name="Court" type="s:ST_String"/>
                <xsd:element name="Day" type="s:ST_String"/>
                <xsd:element name="DayAccessed" type="s:ST_String"/>
                <xsd:element name="Department" type="s:ST_String"/>
                <xsd:element name="Distributor" type="s:ST_String"/>
                <xsd:element name="Edition" type="s:ST_String"/>
                <xsd:element name="Guid" type="s:ST_String"/>
                <xsd:element name="Institution" type="s:ST_String"/>
                <xsd:element name="InternetSiteTitle" type="s:ST_String"/>
                <xsd:element name="Issue" type="s:ST_String"/>
                <xsd:element name="JournalName" type="s:ST_String"/>
                <xsd:element name="LCID" type="s:ST_Lang"/>
                <xsd:element name="Medium" type="s:ST_String"/>
                <xsd:element name="Month" type="s:ST_String"/>
                <xsd:element name="MonthAccessed" type="s:ST_String"/>
                <xsd:element name="NumberVolumes" type="s:ST_String"/>
                <xsd:element name="Pages" type="s:ST_String"/>
                <xsd:element name="PatentNumber" type="s:ST_String"/>
                <xsd:element name="PeriodicalTitle" type="s:ST_String"/>
                <xsd:element name="ProductionCompany" type="s:ST_String"/>
                <xsd:element name="PublicationTitle" type="s:ST_String"/>
                <xsd:element name="Publisher" type="s:ST_String"/>
                <xsd:element name="RecordingNumber" type="s:ST_String"/>
                <xsd:element name="RefOrder" type="s:ST_String"/>
                <xsd:element name="Reporter" type="s:ST_String"/>
                <xsd:element name="SourceType" type="ST_SourceType"/>
                <xsd:element name="ShortTitle" type="s:ST_String"/>
                <xsd:element name="StandardNumber" type="s:ST_String"/>
                <xsd:element name="StateProvince" type="s:ST_String"/>
                <xsd:element name="Station" type="s:ST_String"/>
                <xsd:element name="Tag" type="s:ST_String"/>
                <xsd:element name="Theater" type="s:ST_String"/>
                <xsd:element name="ThesisType" type="s:ST_String"/>
                <xsd:element name="Title" type="s:ST_String"/>
                <xsd:element name="Type" type="s:ST_String"/>
                <xsd:element name="URL" type="s:ST_String"/>
                <xsd:element name="Version" type="s:ST_String"/>
                <xsd:element name="Volume" type="s:ST_String"/>
                <xsd:element name="Year" type="s:ST_String"/>
                <xsd:element name="YearAccessed" type="s:ST_String"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="Sources" type="CT_Sources"/>
    <xsd:complexType name="CT_Sources">
        <xsd:sequence>
            <xsd:element name="Source" type="CT_SourceType" minOccurs="0" maxOccurs="unbounded"/>
```

```
        </xsd:sequence>
        <xsd:attribute name="SelectedStyle" type="s:ST_String"/>
        <xsd:attribute name="StyleName" type="s:ST_String"/>
        <xsd:attribute name="URI" type="s:ST_String"/>
    </xsd:complexType>
</xsd:schema>
```

A.7.7 Additional Characteristics

This schema is available in the file shared-additionalCharacteristics.xsd.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.openxmlformats.org/officeDocument/2006/characteristics"
  targetNamespace="http://schemas.openxmlformats.org/officeDocument/2006/characteristics"
  elementFormDefault="qualified">
    <xsd:complexType name="CT_AdditionalCharacteristics">
        <xsd:sequence>
            <xsd:element name="characteristic" type="CT_Characteristic" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Characteristic">
        <xsd:attribute name="name" type="xsd:string" use="required"/>
        <xsd:attribute name="relation" type="ST_Relation" use="required"/>
        <xsd:attribute name="val" type="xsd:string" use="required"/>
        <xsd:attribute name="vocabulary" type="xsd:anyURI" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Relation">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="ge"/>
            <xsd:enumeration value="le"/>
            <xsd:enumeration value="gt"/>
            <xsd:enumeration value="lt"/>
            <xsd:enumeration value="eq"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="additionalCharacteristics" type="CT_AdditionalCharacteristics"/>
</xsd:schema>
```

A.7.8 Office Document Relationships

This schema is available in the file shared-relationshipReference.xsd.

```
<xsd:schema xmlns="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
  xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
  targetNamespace="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
  blockDefault="#all">
    <xsd:simpleType name="ST_RelationshipId">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:attribute name="id" type="ST_RelationshipId"/>
    <xsd:attribute name="embed" type="ST_RelationshipId"/>
```



```
    <xsd:attribute name="link" type="ST_RelationshipId"/>
    <xsd:attribute name="dm" type="ST_RelationshipId" default=""/>
    <xsd:attribute name="lo" type="ST_RelationshipId" default=""/>
    <xsd:attribute name="qs" type="ST_RelationshipId" default=""/>
    <xsd:attribute name="cs" type="ST_RelationshipId" default=""/>
    <xsd:attribute name="blip" type="ST_RelationshipId" default=""/>
    <xsd:attribute name="pict" type="ST_RelationshipId"/>
    <xsd:attribute name="href" type="ST_RelationshipId"/>
    <xsd:attribute name="topLeft" type="ST_RelationshipId"/>
    <xsd:attribute name="topRight" type="ST_RelationshipId"/>
    <xsd:attribute name="bottomLeft" type="ST_RelationshipId"/>
    <xsd:attribute name="bottomRight" type="ST_RelationshipId"/>
</xsd:schema>
```

A.7.9 Shared Simple Types

This schema is available in the file shared-commonSimpleTypes.xsd.

```
<xsd:schema xmlns="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
  elementFormDefault="qualified">
    <xsd:simpleType name="ST_Lang">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_HexColorRGB">
        <xsd:restriction base="xsd:hexBinary">
            <xsd:length value="3" fixed="true"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Panose">
        <xsd:restriction base="xsd:hexBinary">
            <xsd:length value="10"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_CalendarType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="gregorian"/>
            <xsd:enumeration value="gregorianUs"/>
            <xsd:enumeration value="gregorianMeFrench"/>
            <xsd:enumeration value="gregorianArabic"/>
            <xsd:enumeration value="hijri"/>
            <xsd:enumeration value="hebrew"/>
            <xsd:enumeration value="taiwan"/>
            <xsd:enumeration value="japan"/>
            <xsd:enumeration value="thai"/>
            <xsd:enumeration value="korea"/>
            <xsd:enumeration value="saka"/>
            <xsd:enumeration value="gregorianXlitEnglish"/>
            <xsd:enumeration value="gregorianXlitFrench"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
```

```
    <xsd:simpleType name="ST_AlgClass">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="hash"/>
            <xsd:enumeration value="custom"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_CryptProv">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="rsaAES"/>
            <xsd:enumeration value="rsaFull"/>
            <xsd:enumeration value="custom"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_AlgType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="typeAny"/>
            <xsd:enumeration value="custom"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ColorType">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Guid">
        <xsd:restriction base="xsd:token">
            <xsd:pattern value="\{[0-9A-F]{8}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{12}\}"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_OnOff">
        <xsd:union memberTypes="xsd:boolean ST_OnOff1"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_OnOff1">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="on"/>
            <xsd:enumeration value="off"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_String">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_XmlName">
        <xsd:restriction base="xsd:NCName">
            <xsd:minLength value="1"/>
            <xsd:maxLength value="255"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TrueFalse">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="t"/>
            <xsd:enumeration value="f"/>
            <xsd:enumeration value="true"/>
            <xsd:enumeration value="false"/>
        </xsd:restriction>
    </xsd:simpleType>
```



```
    <xsd:simpleType name="ST_TrueFalseBlank">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="t"/>
            <xsd:enumeration value="f"/>
            <xsd:enumeration value="true"/>
            <xsd:enumeration value="false"/>
            <xsd:enumeration value=""/>
            <xsd:enumeration value="True"/>
            <xsd:enumeration value="False"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_UnsignedDecimalNumber">
        <xsd:restriction base="xsd:unsignedLong"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TwipsMeasure">
        <xsd:union memberTypes="ST_UnsignedDecimalNumber ST_PositiveUniversalMeasure"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_VerticalAlignRun">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="baseline"/>
            <xsd:enumeration value="superscript"/>
            <xsd:enumeration value="subscript"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Xstring">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_XAlign">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="right"/>
            <xsd:enumeration value="inside"/>
            <xsd:enumeration value="outside"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_YAlign">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="inline"/>
            <xsd:enumeration value="top"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="bottom"/>
            <xsd:enumeration value="inside"/>
            <xsd:enumeration value="outside"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ConformanceClass">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="strict"/>
            <xsd:enumeration value="transitional"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_UniversalMeasure">
```

```
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="-?[0-9]+(\.[0-9]+)?(mm|cm|in|pt|pc|pi)"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PositiveUniversalMeasure">
        <xsd:restriction base="ST_UniversalMeasure">
            <xsd:pattern value="[0-9]+(\.[0-9]+)?(mm|cm|in|pt|pc|pi)"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Percentage">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="-?[0-9]+(\.[0-9]+)?%"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FixedPercentage">
        <xsd:restriction base="ST_Percentage">
            <xsd:pattern value="-?((100)|([0-9][0-9]?))(\.[0-9][0-9]?)?%"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PositivePercentage">
        <xsd:restriction base="ST_Percentage">
            <xsd:pattern value="[0-9]+(\.[0-9]+)?%"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PositiveFixedPercentage">
        <xsd:restriction base="ST_Percentage">
            <xsd:pattern value="((100)|([0-9][0-9]?))(\.[0-9][0-9]?)?%"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>
```

A.8 Custom XML Schema References

This schema is available in the file shared-customXmlSchemaProperties.xsd.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.openxmlformats.org/schemaLibrary/2006/main"
  targetNamespace="http://schemas.openxmlformats.org/schemaLibrary/2006/main"
  attributeFormDefault="qualified" elementFormDefault="qualified">
    <xsd:complexType name="CT_Schema">
        <xsd:attribute name="uri" type="xsd:string" default=""/>
        <xsd:attribute name="manifestLocation" type="xsd:string"/>
        <xsd:attribute name="schemaLocation" type="xsd:string"/>
        <xsd:attribute name="schemaLanguage" type="xsd:token"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SchemaLibrary">
        <xsd:sequence>
            <xsd:element name="schema" type="CT_Schema" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="schemaLibrary" type="CT_SchemaLibrary"/>
</xsd:schema>
```


Annex B. (informative) Schemas – RELAX NG

This annex is informative.

This Office Open XML specification includes a family of schemas defined using the RELAX NG syntax. The definitions of these schemas follow below, and they also reside in an accompanying file named OfficeOpenXML-RELAXNG-Transitional.zip, which is distributed in electronic form.

As well as the differences between RELAX NG and XML Schemas described in Part 1, §B, "Schemas – RELAX NG", here are some other differences:

- The RELAX NG schemas represent co-occurrence constraints between elements and attributes. For example, pml.rnc specifies that the pic element and the attribute spid in p_CT_OleObject are mutually exclusive. Meanwhile, pml.xsd simply allows both in CT_OleObject.
- VML drawing parts (§8.1) can be validated against RELAX NG schemas, but cannot be validated against XSD schemas. This is because there are no XSD schemas for the unqualified xml element, which is the root element of VML drawing parts.

B.1 WordprocessingML

This schema is available in the file wml.rnc.

```
namespace m =
  "http://schemas.openxmlformats.org/officeDocument/2006/math"
namespace o = "urn:schemas-microsoft-com:office:office"
namespace r =
  "http://schemas.openxmlformats.org/officeDocument/2006/relationships"
namespace s =
  "http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
namespace sl =
  "http://schemas.openxmlformats.org/schemaLibrary/2006/main"
namespace v = "urn:schemas-microsoft-com:vml"
default namespace w =
  "http://schemas.openxmlformats.org/wordprocessingml/2006/main"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace wp =
  "http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing"
namespace x = "urn:schemas-microsoft-com:office:excel"

w_CT_Empty = empty
w_CT_OnOff = attribute w:val { s_ST_OnOff }?
```

```
w_ST_LongHexNumber = xsd:hexBinary { length = "4" }
w_CT_LongHexNumber = attribute w:val { w_ST_LongHexNumber }
w_ST_ShortHexNumber = xsd:hexBinary { length = "2" }
w_ST_UcharHexNumber = xsd:hexBinary { length = "1" }
w_CT_Charset =
  attribute w:val { w_ST_UcharHexNumber }?,
  attribute w:characterSet { s_ST_String }?
w_ST_DecimalNumberOrPercent =
  w_ST_UnqualifiedPercentage | s_ST_Percentage
w_ST_UnqualifiedPercentage = xsd:integer
w_ST_DecimalNumber = xsd:integer
w_CT_DecimalNumber = attribute w:val { w_ST_DecimalNumber }
w_CT_UnsignedDecimalNumber =
  attribute w:val { s_ST_UnsignedDecimalNumber }
w_CT_DecimalNumberOrPrecent =
  attribute w:val { w_ST_DecimalNumberOrPercent }
w_CT_TwipsMeasure = attribute w:val { s_ST_TwipsMeasure }
w_ST_SignedTwipsMeasure = xsd:integer | s_ST_UniversalMeasure
w_CT_SignedTwipsMeasure = attribute w:val { w_ST_SignedTwipsMeasure }
w_ST_PixelsMeasure = s_ST_UnsignedDecimalNumber
w_CT_PixelsMeasure = attribute w:val { w_ST_PixelsMeasure }
w_ST_HpsMeasure =
  s_ST_UnsignedDecimalNumber | s_ST_PositiveUniversalMeasure
w_CT_HpsMeasure = attribute w:val { w_ST_HpsMeasure }
w_ST_SignedHpsMeasure = xsd:integer | s_ST_UniversalMeasure
w_CT_SignedHpsMeasure = attribute w:val { w_ST_SignedHpsMeasure }
w_ST_DateTime = xsd:dateTime
w_ST_MacroName = xsd:string { maxLength = "33" }
w_CT_MacroName = attribute w:val { w_ST_MacroName }
w_ST_EighthPointMeasure = s_ST_UnsignedDecimalNumber
w_ST_PointMeasure = s_ST_UnsignedDecimalNumber
w_CT_String = attribute w:val { s_ST_String }
w_ST_TextScale = w_ST_TextScalePercent | w_ST_TextScaleDecimal
w_ST_TextScalePercent = xsd:string { pattern = "0*(600|([0-5]?[0-9]?[0-9]))%" }
w_ST_TextScaleDecimal = xsd:integer { minInclusive = "0" maxInclusive = "600" }
w_CT_TextScale = attribute w:val { w_ST_TextScale }?
w_ST_HighlightColor =
  string "black"
  | string "blue"
  | string "cyan"
  | string "green"
  | string "magenta"
  | string "red"
  | string "yellow"
  | string "white"
  | string "darkBlue"
  | string "darkCyan"
  | string "darkGreen"
  | string "darkMagenta"
  | string "darkRed"
  | string "darkYellow"
  | string "darkGray"
  | string "lightGray"
```



```
  | string "none"
w_CT_Highlight = attribute w:val { w_ST_HighlightColor }
w_ST_HexColorAuto = string "auto"
w_ST_HexColor = w_ST_HexColorAuto | s_ST_HexColorRGB
w_CT_Color =
  attribute w:val { w_ST_HexColor },
  attribute w:themeColor { w_ST_ThemeColor }?,
  attribute w:themeTint { w_ST_UcharHexNumber }?,
  attribute w:themeShade { w_ST_UcharHexNumber }?
w_CT_Lang = attribute w:val { s_ST_Lang }
w_CT_Guid = attribute w:val { s_ST_Guid }?
w_ST_Underline =
  string "single"
  | string "words"
  | string "double"
  | string "thick"
  | string "dotted"
  | string "dottedHeavy"
  | string "dash"
  | string "dashedHeavy"
  | string "dashLong"
  | string "dashLongHeavy"
  | string "dotDash"
  | string "dashDotHeavy"
  | string "dotDotDash"
  | string "dashDotDotHeavy"
  | string "wave"
  | string "wavyHeavy"
  | string "wavyDouble"
  | string "none"
w_CT_Underline =
  attribute w:val { w_ST_Underline }?,
  attribute w:color { w_ST_HexColor }?,
  attribute w:themeColor { w_ST_ThemeColor }?,
  attribute w:themeTint { w_ST_UcharHexNumber }?,
  attribute w:themeShade { w_ST_UcharHexNumber }?
w_ST_TextEffect =
  string "blinkBackground"
  | string "lights"
  | string "antsBlack"
  | string "antsRed"
  | string "shimmer"
  | string "sparkle"
  | string "none"
w_CT_TextEffect = attribute w:val { w_ST_TextEffect }
w_ST_Border =
  string "nil"
  | string "none"
  | string "single"
  | string "thick"
  | string "double"
  | string "dotted"
  | string "dashed"
```

```
    | string "dotDash"
    | string "dotDotDash"
    | string "triple"
    | string "thinThickSmallGap"
    | string "thickThinSmallGap"
    | string "thinThickThinSmallGap"
    | string "thinThickMediumGap"
    | string "thickThinMediumGap"
    | string "thinThickThinMediumGap"
    | string "thinThickLargeGap"
    | string "thickThinLargeGap"
    | string "thinThickThinLargeGap"
    | string "wave"
    | string "doubleWave"
    | string "dashSmallGap"
    | string "dashDotStroked"
    | string "threeDEmboss"
    | string "threeDEngrave"
    | string "outset"
    | string "inset"
    | string "apples"
    | string "archedScallops"
    | string "babyPacifier"
    | string "babyRattle"
    | string "balloons3Colors"
    | string "balloonsHotAir"
    | string "basicBlackDashes"
    | string "basicBlackDots"
    | string "basicBlackSquares"
    | string "basicThinLines"
    | string "basicWhiteDashes"
    | string "basicWhiteDots"
    | string "basicWhiteSquares"
    | string "basicWideInline"
    | string "basicWideMidline"
    | string "basicWideOutline"
    | string "bats"
    | string "birds"
    | string "birdsFlight"
    | string "cabins"
    | string "cakeSlice"
    | string "candyCorn"
    | string "celticKnotwork"
    | string "certificateBanner"
    | string "chainLink"
    | string "champagneBottle"
    | string "checkedBarBlack"
    | string "checkedBarColor"
    | string "checkered"
    | string "christmasTree"
    | string "circlesLines"
    | string "circlesRectangles"
    | string "classicalWave"
```



```
    | string "clocks"
    | string "compass"
    | string "confetti"
    | string "confettiGrays"
    | string "confettiOutline"
    | string "confettiStreamers"
    | string "confettiWhite"
    | string "cornerTriangles"
    | string "couponCutoutDashes"
    | string "couponCutoutDots"
    | string "crazyMaze"
    | string "creaturesButterfly"
    | string "creaturesFish"
    | string "creaturesInsects"
    | string "creaturesLadyBug"
    | string "crossStitch"
    | string "cup"
    | string "decoArch"
    | string "decoArchColor"
    | string "decoBlocks"
    | string "diamondsGray"
    | string "doubleD"
    | string "doubleDiamonds"
    | string "earth1"
    | string "earth2"
    | string "earth3"
    | string "eclipsingSquares1"
    | string "eclipsingSquares2"
    | string "eggsBlack"
    | string "fans"
    | string "film"
    | string "firecrackers"
    | string "flowersBlockPrint"
    | string "flowersDaisies"
    | string "flowersModern1"
    | string "flowersModern2"
    | string "flowersPansy"
    | string "flowersRedRose"
    | string "flowersRoses"
    | string "flowersTeacup"
    | string "flowersTiny"
    | string "gems"
    | string "gingerbreadMan"
    | string "gradient"
    | string "handmade1"
    | string "handmade2"
    | string "heartBalloon"
    | string "heartGray"
    | string "hearts"
    | string "heebieJeebies"
    | string "holly"
    | string "houseFunky"
    | string "hypnotic"
```

```
    | string "iceCreamCones"
    | string "lightBulb"
    | string "lightning1"
    | string "lightning2"
    | string "mapPins"
    | string "mapleLeaf"
    | string "mapleMuffins"
    | string "marquee"
    | string "marqueeToothed"
    | string "moons"
    | string "mosaic"
    | string "musicNotes"
    | string "northwest"
    | string "ovals"
    | string "packages"
    | string "palmsBlack"
    | string "palmsColor"
    | string "paperClips"
    | string "papyrus"
    | string "partyFavor"
    | string "partyGlass"
    | string "pencils"
    | string "people"
    | string "peopleWaving"
    | string "peopleHats"
    | string "poinsettias"
    | string "postageStamp"
    | string "pumpkin1"
    | string "pushPinNote2"
    | string "pushPinNote1"
    | string "pyramids"
    | string "pyramidsAbove"
    | string "quadrants"
    | string "rings"
    | string "safari"
    | string "sawtooth"
    | string "sawtoothGray"
    | string "scaredCat"
    | string "seattle"
    | string "shadowedSquares"
    | string "sharksTeeth"
    | string "shorebirdTracks"
    | string "skyrocket"
    | string "snowflakeFancy"
    | string "snowflakes"
    | string "sombrero"
    | string "southwest"
    | string "stars"
    | string "starsTop"
    | string "stars3d"
    | string "starsBlack"
    | string "starsShadowed"
    | string "sun"
```



```
  | string "swirligig"
  | string "tornPaper"
  | string "tornPaperBlack"
  | string "trees"
  | string "triangleParty"
  | string "triangles"
  | string "triangle1"
  | string "triangle2"
  | string "triangleCircle1"
  | string "triangleCircle2"
  | string "shapes1"
  | string "shapes2"
  | string "twistedLines1"
  | string "twistedLines2"
  | string "vine"
  | string "waveline"
  | string "weavingAngles"
  | string "weavingBraid"
  | string "weavingRibbon"
  | string "weavingStrips"
  | string "whiteFlowers"
  | string "woodwork"
  | string "xIllusions"
  | string "zanyTriangles"
  | string "zigZag"
  | string "zigZagStitch"
  | string "custom"
w_CT_Border =
  attribute w:val { w_ST_Border },
  attribute w:color { w_ST_HexColor }?,
  attribute w:themeColor { w_ST_ThemeColor }?,
  attribute w:themeTint { w_ST_UcharHexNumber }?,
  attribute w:themeShade { w_ST_UcharHexNumber }?,
  attribute w:sz { w_ST_EighthPointMeasure }?,
  attribute w:space { w_ST_PointMeasure }?,
  attribute w:shadow { s_ST_OnOff }?,
  attribute w:frame { s_ST_OnOff }?
w_ST_Shd =
  string "nil"
  | string "clear"
  | string "solid"
  | string "horzStripe"
  | string "vertStripe"
  | string "reverseDiagStripe"
  | string "diagStripe"
  | string "horzCross"
  | string "diagCross"
  | string "thinHorzStripe"
  | string "thinVertStripe"
  | string "thinReverseDiagStripe"
  | string "thinDiagStripe"
  | string "thinHorzCross"
  | string "thinDiagCross"
```

```
  | string "pct5"
  | string "pct10"
  | string "pct12"
  | string "pct15"
  | string "pct20"
  | string "pct25"
  | string "pct30"
  | string "pct35"
  | string "pct37"
  | string "pct40"
  | string "pct45"
  | string "pct50"
  | string "pct55"
  | string "pct60"
  | string "pct62"
  | string "pct65"
  | string "pct70"
  | string "pct75"
  | string "pct80"
  | string "pct85"
  | string "pct87"
  | string "pct90"
  | string "pct95"
w_CT_Shd =
  attribute w:val { w_ST_Shd },
  attribute w:color { w_ST_HexColor }?,
  attribute w:themeColor { w_ST_ThemeColor }?,
  attribute w:themeTint { w_ST_UcharHexNumber }?,
  attribute w:themeShade { w_ST_UcharHexNumber }?,
  attribute w:fill { w_ST_HexColor }?,
  attribute w:themeFill { w_ST_ThemeColor }?,
  attribute w:themeFillTint { w_ST_UcharHexNumber }?,
  attribute w:themeFillShade { w_ST_UcharHexNumber }?
w_CT_VerticalAlignRun = attribute w:val { s_ST_VerticalAlignRun }
w_CT_FitText =
  attribute w:val { s_ST_TwipsMeasure },
  attribute w:id { w_ST_DecimalNumber }?
w_ST_Em =
  string "none"
  | string "dot"
  | string "comma"
  | string "circle"
  | string "underDot"
w_CT_Em = attribute w:val { w_ST_Em }
w_CT_Language =
  attribute w:val { s_ST_Lang }?,
  attribute w:eastAsia { s_ST_Lang }?,
  attribute w:bidi { s_ST_Lang }?
w_ST_CombineBrackets =
  string "none"
  | string "round"
  | string "square"
  | string "angle"
```



```
  | string "curly"
w_CT_EastAsianLayout =
  attribute w:id { w_ST_DecimalNumber }?,
  attribute w:combine { s_ST_OnOff }?,
  attribute w:combineBrackets { w_ST_CombineBrackets }?,
  attribute w:vert { s_ST_OnOff }?,
  attribute w:vertCompress { s_ST_OnOff }?
w_ST_HeightRule = string "auto" | string "exact" | string "atLeast"
w_ST_Wrap =
  string "auto"
  | string "notBeside"
  | string "around"
  | string "tight"
  | string "through"
  | string "none"
w_ST_VAnchor = string "text" | string "margin" | string "page"
w_ST_HAnchor = string "text" | string "margin" | string "page"
w_ST_DropCap = string "none" | string "drop" | string "margin"
w_CT_FramePr =
  attribute w:dropCap { w_ST_DropCap }?,
  attribute w:lines { w_ST_DecimalNumber }?,
  attribute w:w { s_ST_TwipsMeasure }?,
  attribute w:h { s_ST_TwipsMeasure }?,
  attribute w:vSpace { s_ST_TwipsMeasure }?,
  attribute w:hSpace { s_ST_TwipsMeasure }?,
  attribute w:wrap { w_ST_Wrap }?,
  attribute w:hAnchor { w_ST_HAnchor }?,
  attribute w:vAnchor { w_ST_VAnchor }?,
  attribute w:x { w_ST_SignedTwipsMeasure }?,
  attribute w:xAlign { s_ST_XAlign }?,
  attribute w:y { w_ST_SignedTwipsMeasure }?,
  attribute w:yAlign { s_ST_YAlign }?,
  attribute w:hRule { w_ST_HeightRule }?,
  attribute w:anchorLock { s_ST_OnOff }?
w_ST_TabJc =
  string "clear"
  | string "start"
  | string "center"
  | string "end"
  | string "decimal"
  | string "bar"
  | string "num"
  | string "left"
  | string "right"
w_ST_TabTlc =
  string "none"
  | string "dot"
  | string "hyphen"
  | string "underscore"
  | string "heavy"
  | string "middleDot"
w_CT_TabStop =
  attribute w:val { w_ST_TabJc },
```

```
  attribute w:leader { w_ST_TabTlc }?,
  attribute w:pos { w_ST_SignedTwipsMeasure }
w_ST_LineSpacingRule = string "auto" | string "exact" | string "atLeast"
w_CT_Spacing =
  attribute w:before { s_ST_TwipsMeasure }?,
  attribute w:beforeLines { w_ST_DecimalNumber }?,
  attribute w:beforeAutospacing { s_ST_OnOff }?,
  attribute w:after { s_ST_TwipsMeasure }?,
  attribute w:afterLines { w_ST_DecimalNumber }?,
  attribute w:afterAutospacing { s_ST_OnOff }?,
  attribute w:line { w_ST_SignedTwipsMeasure }?,
  attribute w:lineRule { w_ST_LineSpacingRule }?
w_CT_Ind =
  attribute w:start { w_ST_SignedTwipsMeasure }?,
  attribute w:startChars { w_ST_DecimalNumber }?,
  attribute w:end { w_ST_SignedTwipsMeasure }?,
  attribute w:endChars { w_ST_DecimalNumber }?,
  attribute w:left { w_ST_SignedTwipsMeasure }?,
  attribute w:leftChars { w_ST_DecimalNumber }?,
  attribute w:right { w_ST_SignedTwipsMeasure }?,
  attribute w:rightChars { w_ST_DecimalNumber }?,
  attribute w:hanging { s_ST_TwipsMeasure }?,
  attribute w:hangingChars { w_ST_DecimalNumber }?,
  attribute w:firstLine { s_ST_TwipsMeasure }?,
  attribute w:firstLineChars { w_ST_DecimalNumber }?
w_ST_Jc =
  string "start"
  | string "center"
  | string "end"
  | string "both"
  | string "mediumKashida"
  | string "distribute"
  | string "numTab"
  | string "highKashida"
  | string "lowKashida"
  | string "thaiDistribute"
  | string "left"
  | string "right"
w_ST_JcTable =
  string "center"
  | string "end"
  | string "left"
  | string "right"
  | string "start"
w_CT_Jc = attribute w:val { w_ST_Jc }
w_CT_JcTable = attribute w:val { w_ST_JcTable }
w_ST_View =
  string "none"
  | string "print"
  | string "outline"
  | string "masterPages"
  | string "normal"
  | string "web"
```



```
w_CT_View = attribute w:val { w_ST_View }
w_ST_Zoom =
  string "none"
  | string "fullPage"
  | string "bestFit"
  | string "textFit"
w_CT_Zoom =
  attribute w:val { w_ST_Zoom }?,
  attribute w:percent { w_ST_DecimalNumberOrPercent }
w_CT_WritingStyle =
  attribute w:lang { s_ST_Lang },
  attribute w:vendorID { s_ST_String },
  attribute w:dllVersion { s_ST_String },
  attribute w:nlCheck { s_ST_OnOff }?,
  attribute w:checkStyle { s_ST_OnOff },
  attribute w:appName { s_ST_String }
w_ST_Proof = string "clean" | string "dirty"
w_CT_Proof =
  attribute w:spelling { w_ST_Proof }?,
  attribute w:grammar { w_ST_Proof }?
w_ST_DocType = xsd:string
w_CT_DocType = attribute w:val { w_ST_DocType }
w_ST_DocProtect =
  string "none"
  | string "readOnly"
  | string "comments"
  | string "trackedChanges"
  | string "forms"
w_AG_Password =
  attribute w:algorithmName { s_ST_String }?,
  attribute w:hashValue { xsd:base64Binary }?,
  attribute w:saltValue { xsd:base64Binary }?,
  attribute w:spinCount { w_ST_DecimalNumber }?
w_AG_TransitionalPassword =
  attribute w:cryptProviderType { s_ST_CryptProv }?,
  attribute w:cryptAlgorithmClass { s_ST_AlgClass }?,
  attribute w:cryptAlgorithmType { s_ST_AlgType }?,
  attribute w:cryptAlgorithmSid { w_ST_DecimalNumber }?,
  attribute w:cryptSpinCount { w_ST_DecimalNumber }?,
  attribute w:cryptProvider { s_ST_String }?,
  attribute w:algIdExt { w_ST_LongHexNumber }?,
  attribute w:algIdExtSource { s_ST_String }?,
  attribute w:cryptProviderTypeExt { w_ST_LongHexNumber }?,
  attribute w:cryptProviderTypeExtSource { s_ST_String }?,
  attribute w:hash { xsd:base64Binary }?,
  attribute w:salt { xsd:base64Binary }?
w_CT_DocProtect =
  attribute w:edit { w_ST_DocProtect }?,
  attribute w:formatting { s_ST_OnOff }?,
  attribute w:enforcement { s_ST_OnOff }?,
  w_AG_Password,
  w_AG_TransitionalPassword
w_ST_MailMergeDocType =
```

```
  string "catalog"
  | string "envelopes"
  | string "mailingLabels"
  | string "formLetters"
  | string "email"
  | string "fax"
w_CT_MailMergeDocType = attribute w:val { w_ST_MailMergeDocType }
w_ST_MailMergeDataType = xsd:string
w_CT_MailMergeDataType = attribute w:val { w_ST_MailMergeDataType }
w_ST_MailMergeDest =
  string "newDocument"
  | string "printer"
  | string "email"
  | string "fax"
w_CT_MailMergeDest = attribute w:val { w_ST_MailMergeDest }
w_ST_MailMergeOdsoFMDFieldType = string "null" | string "dbColumn"
w_CT_MailMergeOdsoFMDFieldType =
  attribute w:val { w_ST_MailMergeOdsoFMDFieldType }
w_CT_TrackChangesView =
  attribute w:markup { s_ST_OnOff }?,
  attribute w:comments { s_ST_OnOff }?,
  attribute w:insDel { s_ST_OnOff }?,
  attribute w:formatting { s_ST_OnOff }?,
  attribute w:inkAnnotations { s_ST_OnOff }?
w_CT_Kinsoku =
  attribute w:lang { s_ST_Lang },
  attribute w:val { s_ST_String }
w_ST_TextDirection =
  string "tb"
  | string "rl"
  | string "lr"
  | string "tbV"
  | string "rlV"
  | string "lrV"
  | string "btLr"
  | string "lrTb"
  | string "lrTbV"
  | string "tbLrV"
  | string "tbRl"
  | string "tbRlV"
w_CT_TextDirection = attribute w:val { w_ST_TextDirection }
w_ST_TextAlignment =
  string "top"
  | string "center"
  | string "baseline"
  | string "bottom"
  | string "auto"
w_CT_TextAlignment = attribute w:val { w_ST_TextAlignment }
w_ST_DisplacedByCustomXml = string "next" | string "prev"
w_ST_AnnotationVMerge = string "cont" | string "rest"
w_CT_Markup = attribute w:id { w_ST_DecimalNumber }
w_CT_TrackChange =
  w_CT_Markup,
```



```
  attribute w:author { s_ST_String },
  attribute w:date { w_ST_DateTime }?
w_CT_CellMergeTrackChange =
  w_CT_TrackChange,
  attribute w:vMerge { w_ST_AnnotationVMerge }?,
  attribute w:vMergeOrig { w_ST_AnnotationVMerge }?
w_CT_TrackChangeRange =
  w_CT_TrackChange,
  attribute w:displacedByCustomXml { w_ST_DisplacedByCustomXml }?
w_CT_MarkupRange =
  w_CT_Markup,
  attribute w:displacedByCustomXml { w_ST_DisplacedByCustomXml }?
w_CT_BookmarkRange =
  w_CT_MarkupRange,
  attribute w:colFirst { w_ST_DecimalNumber }?,
  attribute w:colLast { w_ST_DecimalNumber }?
w_CT_Bookmark =
  w_CT_BookmarkRange,
  attribute w:name { s_ST_String }
w_CT_MoveBookmark =
  w_CT_Bookmark,
  attribute w:author { s_ST_String },
  attribute w:date { w_ST_DateTime }
w_CT_Comment =
  w_CT_TrackChange,
  w_EG_BlockLevelElts*,
  attribute w:initials { s_ST_String }?
w_CT_TrackChangeNumbering =
  w_CT_TrackChange,
  attribute w:original { s_ST_String }?
w_CT_TblPrExChange =
  w_CT_TrackChange,
  element tblPrEx { w_CT_TblPrExBase }
w_CT_TcPrChange =
  w_CT_TrackChange,
  element tcPr { w_CT_TcPrInner }
w_CT_TrPrChange =
  w_CT_TrackChange,
  element trPr { w_CT_TrPrBase }
w_CT_TblGridChange =
  w_CT_Markup,
  element tblGrid { w_CT_TblGridBase }
w_CT_TblPrChange =
  w_CT_TrackChange,
  element tblPr { w_CT_TblPrBase }
w_CT_SectPrChange =
  w_CT_TrackChange,
  element sectPr { w_CT_SectPrBase }?
w_CT_PPrChange =
  w_CT_TrackChange,
  element pPr { w_CT_PPrBase }
w_CT_RPrChange =
  w_CT_TrackChange,
```

```
  element rPr { w_CT_RPrOriginal }
w_CT_ParaRPrChange =
  w_CT_TrackChange,
  element rPr { w_CT_ParaRPrOriginal }
w_CT_RunTrackChange =
  w_CT_TrackChange, (w_EG_ContentRunContent | m_EG_OMathMathElements)*
w_EG_PContentMath = w_EG_PContentBase* | w_EG_ContentRunContentBase*
w_EG_PContentBase =
  element customXml { w_CT_CustomXmlRun }
  | element fldSimple { w_CT_SimpleField }*
  | element hyperlink { w_CT_Hyperlink }
w_EG_ContentRunContentBase =
  element smartTag { w_CT_SmartTagRun }
  | element sdt { w_CT_SdtRun }
  | w_EG_RunLevelElts*
w_EG_CellMarkupElements =
  element cellIns { w_CT_TrackChange }?
  | element cellDel { w_CT_TrackChange }?
  | element cellMerge { w_CT_CellMergeTrackChange }?
w_EG_RangeMarkupElements =
  element bookmarkStart { w_CT_Bookmark }
  | element bookmarkEnd { w_CT_MarkupRange }
  | element moveFromRangeStart { w_CT_MoveBookmark }
  | element moveFromRangeEnd { w_CT_MarkupRange }
  | element moveToRangeStart { w_CT_MoveBookmark }
  | element moveToRangeEnd { w_CT_MarkupRange }
  | element commentRangeStart { w_CT_MarkupRange }
  | element commentRangeEnd { w_CT_MarkupRange }
  | element customXmlInsRangeStart { w_CT_TrackChange }
  | element customXmlInsRangeEnd { w_CT_Markup }
  | element customXmlDelRangeStart { w_CT_TrackChange }
  | element customXmlDelRangeEnd { w_CT_Markup }
  | element customXmlMoveFromRangeStart { w_CT_TrackChange }
  | element customXmlMoveFromRangeEnd { w_CT_Markup }
  | element customXmlMoveToRangeStart { w_CT_TrackChange }
  | element customXmlMoveToRangeEnd { w_CT_Markup }
w_CT_NumPr =
  element ilvl { w_CT_DecimalNumber }?,
  element numId { w_CT_DecimalNumber }?,
  element numberingChange { w_CT_TrackChangeNumbering }?,
  element ins { w_CT_TrackChange }?
w_CT_PBdr =
  element top { w_CT_Border }?,
  element left { w_CT_Border }?,
  element bottom { w_CT_Border }?,
  element right { w_CT_Border }?,
  element between { w_CT_Border }?,
  element bar { w_CT_Border }?
w_CT_Tabs = element tab { w_CT_TabStop }+
w_ST_TextboxTightWrap =
  string "none"
  | string "allLines"
  | string "firstAndLastLine"
```



```
  | string "firstLineOnly"
  | string "lastLineOnly"
w_CT_TextboxTightWrap = attribute w:val { w_ST_TextboxTightWrap }
w_CT_PPr =
  w_CT_PPrBase,
  element rPr { w_CT_ParaRPr }?,
  element sectPr { w_CT_SectPr }?,
  element pPrChange { w_CT_PPrChange }?
w_CT_PPrBase =
  element pStyle { w_CT_String }?,
  element keepNext { w_CT_OnOff }?,
  element keepLines { w_CT_OnOff }?,
  element pageBreakBefore { w_CT_OnOff }?,
  element framePr { w_CT_FramePr }?,
  element widowControl { w_CT_OnOff }?,
  element numPr { w_CT_NumPr }?,
  element suppressLineNumbers { w_CT_OnOff }?,
  element pBdr { w_CT_PBdr }?,
  element shd { w_CT_Shd }?,
  element tabs { w_CT_Tabs }?,
  element suppressAutoHyphens { w_CT_OnOff }?,
  element kinsoku { w_CT_OnOff }?,
  element wordWrap { w_CT_OnOff }?,
  element overflowPunct { w_CT_OnOff }?,
  element topLinePunct { w_CT_OnOff }?,
  element autoSpaceDE { w_CT_OnOff }?,
  element autoSpaceDN { w_CT_OnOff }?,
  element bidi { w_CT_OnOff }?,
  element adjustRightInd { w_CT_OnOff }?,
  element snapToGrid { w_CT_OnOff }?,
  element spacing { w_CT_Spacing }?,
  element ind { w_CT_Ind }?,
  element contextualSpacing { w_CT_OnOff }?,
  element mirrorIndents { w_CT_OnOff }?,
  element suppressOverlap { w_CT_OnOff }?,
  element jc { w_CT_Jc }?,
  element textDirection { w_CT_TextDirection }?,
  element textAlignment { w_CT_TextAlignment }?,
  element textboxTightWrap { w_CT_TextboxTightWrap }?,
  element outlineLvl { w_CT_DecimalNumber }?,
  element divId { w_CT_DecimalNumber }?,
  element cnfStyle { w_CT_Cnf }?
w_CT_PPrGeneral =
  w_CT_PPrBase,
  element pPrChange { w_CT_PPrChange }?
w_CT_Control =
  attribute w:name { s_ST_String }?,
  attribute w:shapeid { s_ST_String }?,
  r_id?
w_CT_Background =
  attribute w:color { w_ST_HexColor }?,
  attribute w:themeColor { w_ST_ThemeColor }?,
  attribute w:themeTint { w_ST_UcharHexNumber }?,
```

```
  attribute w:themeShade { w_ST_UcharHexNumber }?,
  (w_any_vml_vml*, w_any_vml_office*)+,
  element drawing { w_CT_Drawing }?
w_CT_Rel = r_id
w_CT_Object =
  attribute w:dxaOrig { s_ST_TwipsMeasure }?,
  attribute w:dyaOrig { s_ST_TwipsMeasure }?,
  (w_any_vml_vml*, w_any_vml_office*)+,
  element drawing { w_CT_Drawing }?,
  (element control { w_CT_Control }
   | element objectLink { w_CT_ObjectLink }
   | element objectEmbed { w_CT_ObjectEmbed }
   | element movie { w_CT_Rel })?
w_CT_Picture =
  (w_any_vml_vml*, w_any_vml_office*)+,
  element movie { w_CT_Rel }?,
  element control { w_CT_Control }?
w_CT_ObjectEmbed =
  attribute w:drawAspect { w_ST_ObjectDrawAspect }?,
  r_id,
  attribute w:progId { s_ST_String }?,
  attribute w:shapeId { s_ST_String }?,
  attribute w:fieldCodes { s_ST_String }?
w_ST_ObjectDrawAspect = string "content" | string "icon"
w_CT_ObjectLink =
  w_CT_ObjectEmbed,
  attribute w:updateMode { w_ST_ObjectUpdateMode },
  attribute w:lockedField { s_ST_OnOff }?
w_ST_ObjectUpdateMode = string "always" | string "onCall"
w_CT_Drawing = (wp_anchor? | wp_inline?)+
w_CT_SimpleField =
  attribute w:instr { s_ST_String },
  attribute w:fldLock { s_ST_OnOff }?,
  attribute w:dirty { s_ST_OnOff }?,
  element fldData { w_CT_Text }?,
  w_EG_PContent*
w_ST_FldCharType = string "begin" | string "separate" | string "end"
w_ST_InfoTextType = string "text" | string "autoText"
w_ST_FFHelpTextVal = xsd:string { maxLength = "256" }
w_ST_FFStatusTextVal = xsd:string { maxLength = "140" }
w_ST_FFName = xsd:string { maxLength = "65" }
w_ST_FFTextType =
  string "regular"
  | string "number"
  | string "date"
  | string "currentTime"
  | string "currentDate"
  | string "calculated"
w_CT_FFTextType = attribute w:val { w_ST_FFTextType }
w_CT_FFName = attribute w:val { w_ST_FFName }?
w_CT_FldChar =
  attribute w:fldCharType { w_ST_FldCharType },
  attribute w:fldLock { s_ST_OnOff }?,
```



```
  attribute w:dirty { s_ST_OnOff }?,
  (element fldData { w_CT_Text }?
   | element ffData { w_CT_FFData }?
   | element numberingChange { w_CT_TrackChangeNumbering }?)
w_CT_Hyperlink =
  attribute w:tgtFrame { s_ST_String }?,
  attribute w:tooltip { s_ST_String }?,
  attribute w:docLocation { s_ST_String }?,
  attribute w:history { s_ST_OnOff }?,
  attribute w:anchor { s_ST_String }?,
  r_id?,
  w_EG_PContent*
w_CT_FFData =
  (element name { w_CT_FFName }
   | element label { w_CT_DecimalNumber }?
   | element tabIndex { w_CT_UnsignedDecimalNumber }?
   | element enabled { w_CT_OnOff }
   | element calcOnExit { w_CT_OnOff }
   | element entryMacro { w_CT_MacroName }?
   | element exitMacro { w_CT_MacroName }?
   | element helpText { w_CT_FFHelpText }?
   | element statusText { w_CT_FFStatusText }?
   | (element checkBox { w_CT_FFCheckBox }
      | element ddList { w_CT_FFDDList }
      | element textInput { w_CT_FFTextInput }))+
w_CT_FFHelpText =
  attribute w:type { w_ST_InfoTextType }?,
  attribute w:val { w_ST_FFHelpTextVal }?
w_CT_FFStatusText =
  attribute w:type { w_ST_InfoTextType }?,
  attribute w:val { w_ST_FFStatusTextVal }?
w_CT_FFCheckBox =
  (element size { w_CT_HpsMeasure }
   | element sizeAuto { w_CT_OnOff }),
  element default { w_CT_OnOff }?,
  element checked { w_CT_OnOff }?
w_CT_FFDDList =
  element result { w_CT_DecimalNumber }?,
  element default { w_CT_DecimalNumber }?,
  element listEntry { w_CT_String }*
w_CT_FFTextInput =
  element type { w_CT_FFTextType }?,
  element default { w_CT_String }?,
  element maxLength { w_CT_DecimalNumber }?,
  element format { w_CT_String }?
w_ST_SectionMark =
  string "nextPage"
  | string "nextColumn"
  | string "continuous"
  | string "evenPage"
  | string "oddPage"
w_CT_SectType = attribute w:val { w_ST_SectionMark }?
w_CT_PaperSource =
```

```
  attribute w:first { w_ST_DecimalNumber }?,
  attribute w:other { w_ST_DecimalNumber }?
w_ST_NumberFormat =
  string "decimal"
  | string "upperRoman"
  | string "lowerRoman"
  | string "upperLetter"
  | string "lowerLetter"
  | string "ordinal"
  | string "cardinalText"
  | string "ordinalText"
  | string "hex"
  | string "chicago"
  | string "ideographDigital"
  | string "japaneseCounting"
  | string "aiueo"
  | string "iroha"
  | string "decimalFullWidth"
  | string "decimalHalfWidth"
  | string "japaneseLegal"
  | string "japaneseDigitalTenThousand"
  | string "decimalEnclosedCircle"
  | string "decimalFullWidth2"
  | string "aiueoFullWidth"
  | string "irohaFullWidth"
  | string "decimalZero"
  | string "bullet"
  | string "ganada"
  | string "chosung"
  | string "decimalEnclosedFullstop"
  | string "decimalEnclosedParen"
  | string "decimalEnclosedCircleChinese"
  | string "ideographEnclosedCircle"
  | string "ideographTraditional"
  | string "ideographZodiac"
  | string "ideographZodiacTraditional"
  | string "taiwaneseCounting"
  | string "ideographLegalTraditional"
  | string "taiwaneseCountingThousand"
  | string "taiwaneseDigital"
  | string "chineseCounting"
  | string "chineseLegalSimplified"
  | string "chineseCountingThousand"
  | string "koreanDigital"
  | string "koreanCounting"
  | string "koreanLegal"
  | string "koreanDigital2"
  | string "vietnameseCounting"
  | string "russianLower"
  | string "russianUpper"
  | string "none"
  | string "numberInDash"
  | string "hebrew1"
```



```
  | string "hebrew2"
  | string "arabicAlpha"
  | string "arabicAbjad"
  | string "hindiVowels"
  | string "hindiConsonants"
  | string "hindiNumbers"
  | string "hindiCounting"
  | string "thaiLetters"
  | string "thaiNumbers"
  | string "thaiCounting"
  | string "bahtText"
  | string "dollarText"
  | string "custom"
w_ST_PageOrientation = string "portrait" | string "landscape"
w_CT_PageSz =
  attribute w:w { s_ST_TwipsMeasure }?,
  attribute w:h { s_ST_TwipsMeasure }?,
  attribute w:orient { w_ST_PageOrientation }?,
  attribute w:code { w_ST_DecimalNumber }?
w_CT_PageMar =
  attribute w:top { w_ST_SignedTwipsMeasure },
  attribute w:right { s_ST_TwipsMeasure },
  attribute w:bottom { w_ST_SignedTwipsMeasure },
  attribute w:left { s_ST_TwipsMeasure },
  attribute w:header { s_ST_TwipsMeasure },
  attribute w:footer { s_ST_TwipsMeasure },
  attribute w:gutter { s_ST_TwipsMeasure }
w_ST_PageBorderZOrder = string "front" | string "back"
w_ST_PageBorderDisplay =
  string "allPages" | string "firstPage" | string "notFirstPage"
w_ST_PageBorderOffset = string "page" | string "text"
w_CT_PageBorders =
  attribute w:zOrder { w_ST_PageBorderZOrder }?,
  attribute w:display { w_ST_PageBorderDisplay }?,
  attribute w:offsetFrom { w_ST_PageBorderOffset }?,
  element top { w_CT_TopPageBorder }?,
  element left { w_CT_PageBorder }?,
  element bottom { w_CT_BottomPageBorder }?,
  element right { w_CT_PageBorder }?
w_CT_PageBorder = w_CT_Border, r_id?
w_CT_BottomPageBorder = w_CT_PageBorder, r_bottomLeft?, r_bottomRight?
w_CT_TopPageBorder = w_CT_PageBorder, r_topLeft?, r_topRight?
w_ST_ChapterSep =
  string "hyphen"
  | string "period"
  | string "colon"
  | string "emDash"
  | string "enDash"
w_ST_LineNumberRestart =
  string "newPage" | string "newSection" | string "continuous"
w_CT_LineNumber =
  attribute w:countBy { w_ST_DecimalNumber }?,
  attribute w:start { w_ST_DecimalNumber }?,
```

```
  attribute w:distance { s_ST_TwipsMeasure }?,
  attribute w:restart { w_ST_LineNumberRestart }?
w_CT_PageNumber =
  attribute w:fmt { w_ST_NumberFormat }?,
  attribute w:start { w_ST_DecimalNumber }?,
  attribute w:chapStyle { w_ST_DecimalNumber }?,
  attribute w:chapSep { w_ST_ChapterSep }?
w_CT_Column =
  attribute w:w { s_ST_TwipsMeasure }?,
  attribute w:space { s_ST_TwipsMeasure }?
w_CT_Columns =
  attribute w:equalWidth { s_ST_OnOff }?,
  attribute w:space { s_ST_TwipsMeasure }?,
  attribute w:num { w_ST_DecimalNumber }?,
  attribute w:sep { s_ST_OnOff }?,
  element col { w_CT_Column }*
w_ST_VerticalJc =
  string "top" | string "center" | string "both" | string "bottom"
w_CT_VerticalJc = attribute w:val { w_ST_VerticalJc }
w_ST_DocGrid =
  string "default"
  | string "lines"
  | string "linesAndChars"
  | string "snapToChars"
w_CT_DocGrid =
  attribute w:type { w_ST_DocGrid }?,
  attribute w:linePitch { w_ST_DecimalNumber }?,
  attribute w:charSpace { w_ST_DecimalNumber }?
w_ST_HdrFtr = string "even" | string "default" | string "first"
w_ST_FtnEdn =
  string "normal"
  | string "separator"
  | string "continuationSeparator"
  | string "continuationNotice"
w_CT_HdrFtrRef =
  w_CT_Rel,
  attribute w:type { w_ST_HdrFtr }
w_EG_HdrFtrReferences =
  element headerReference { w_CT_HdrFtrRef }?
  | element footerReference { w_CT_HdrFtrRef }?
w_CT_HdrFtr = w_EG_BlockLevelElts+
w_EG_SectPrContents =
  element footnotePr { w_CT_FtnProps }?,
  element endnotePr { w_CT_EdnProps }?,
  element type { w_CT_SectType }?,
  element pgSz { w_CT_PageSz }?,
  element pgMar { w_CT_PageMar }?,
  element paperSrc { w_CT_PaperSource }?,
  element pgBorders { w_CT_PageBorders }?,
  element lnNumType { w_CT_LineNumber }?,
  element pgNumType { w_CT_PageNumber }?,
  element cols { w_CT_Columns }?,
  element formProt { w_CT_OnOff }?,
```



```
  element vAlign { w_CT_VerticalJc }?,
  element noEndnote { w_CT_OnOff }?,
  element titlePg { w_CT_OnOff }?,
  element textDirection { w_CT_TextDirection }?,
  element bidi { w_CT_OnOff }?,
  element rtlGutter { w_CT_OnOff }?,
  element docGrid { w_CT_DocGrid }?,
  element printerSettings { w_CT_Rel }?
w_AG_SectPrAttributes =
  attribute w:rsidRPr { w_ST_LongHexNumber }?,
  attribute w:rsidDel { w_ST_LongHexNumber }?,
  attribute w:rsidR { w_ST_LongHexNumber }?,
  attribute w:rsidSect { w_ST_LongHexNumber }?
w_CT_SectPrBase = w_AG_SectPrAttributes, w_EG_SectPrContents?
w_CT_SectPr =
  w_AG_SectPrAttributes,
  w_EG_HdrFtrReferences*,
  w_EG_SectPrContents?,
  element sectPrChange { w_CT_SectPrChange }?
w_ST_BrType = string "page" | string "column" | string "textWrapping"
w_ST_BrClear =
  string "none" | string "left" | string "right" | string "all"
w_CT_Br =
  attribute w:type { w_ST_BrType }?,
  attribute w:clear { w_ST_BrClear }?
w_ST_PTabAlignment = string "left" | string "center" | string "right"
w_ST_PTabRelativeTo = string "margin" | string "indent"
w_ST_PTabLeader =
  string "none"
  | string "dot"
  | string "hyphen"
  | string "underscore"
  | string "middleDot"
w_CT_PTab =
  attribute w:alignment { w_ST_PTabAlignment },
  attribute w:relativeTo { w_ST_PTabRelativeTo },
  attribute w:leader { w_ST_PTabLeader }
w_CT_Sym =
  attribute w:font { s_ST_String }?,
  attribute w:char { w_ST_ShortHexNumber }?
w_ST_ProofErr =
  string "spellStart"
  | string "spellEnd"
  | string "gramStart"
  | string "gramEnd"
w_CT_ProofErr = attribute w:type { w_ST_ProofErr }
w_ST_EdGrp =
  string "none"
  | string "everyone"
  | string "administrators"
  | string "contributors"
  | string "editors"
  | string "owners"
```

```
  | string "current"
w_CT_Perm =
  attribute w:id { s_ST_String },
  attribute w:displacedByCustomXml { w_ST_DisplacedByCustomXml }?
w_CT_PermStart =
  w_CT_Perm,
  attribute w:edGrp { w_ST_EdGrp }?,
  attribute w:ed { s_ST_String }?,
  attribute w:colFirst { w_ST_DecimalNumber }?,
  attribute w:colLast { w_ST_DecimalNumber }?
w_CT_Text = s_ST_String, xml_space?
w_EG_RunInnerContent =
  element br { w_CT_Br }
  | element t { w_CT_Text }
  | element contentPart { w_CT_Rel }
  | element delText { w_CT_Text }
  | element instrText { w_CT_Text }
  | element delInstrText { w_CT_Text }
  | element noBreakHyphen { w_CT_Empty }
  | element softHyphen { w_CT_Empty }?
  | element dayShort { w_CT_Empty }?
  | element monthShort { w_CT_Empty }?
  | element yearShort { w_CT_Empty }?
  | element dayLong { w_CT_Empty }?
  | element monthLong { w_CT_Empty }?
  | element yearLong { w_CT_Empty }?
  | element annotationRef { w_CT_Empty }?
  | element footnoteRef { w_CT_Empty }?
  | element endnoteRef { w_CT_Empty }?
  | element separator { w_CT_Empty }?
  | element continuationSeparator { w_CT_Empty }?
  | element sym { w_CT_Sym }?
  | element pgNum { w_CT_Empty }?
  | element cr { w_CT_Empty }?
  | element tab { w_CT_Empty }?
  | element object { w_CT_Object }
  | element pict { w_CT_Picture }
  | element fldChar { w_CT_FldChar }
  | element ruby { w_CT_Ruby }
  | element footnoteReference { w_CT_FtnEdnRef }
  | element endnoteReference { w_CT_FtnEdnRef }
  | element commentReference { w_CT_Markup }
  | element drawing { w_CT_Drawing }
  | element ptab { w_CT_PTab }?
  | element lastRenderedPageBreak { w_CT_Empty }?
w_CT_R =
  attribute w:rsidRPr { w_ST_LongHexNumber }?,
  attribute w:rsidDel { w_ST_LongHexNumber }?,
  attribute w:rsidR { w_ST_LongHexNumber }?,
  w_EG_RPr?,
  w_EG_RunInnerContent*
w_ST_Hint = string "default" | string "eastAsia" | string "cs"
w_ST_Theme =
```



```
  string "majorEastAsia"
  | string "majorBidi"
  | string "majorAscii"
  | string "majorHAnsi"
  | string "minorEastAsia"
  | string "minorBidi"
  | string "minorAscii"
  | string "minorHAnsi"
w_CT_Fonts =
  attribute w:hint { w_ST_Hint }?,
  attribute w:ascii { s_ST_String }?,
  attribute w:hAnsi { s_ST_String }?,
  attribute w:eastAsia { s_ST_String }?,
  attribute w:cs { s_ST_String }?,
  attribute w:asciiTheme { w_ST_Theme }?,
  attribute w:hAnsiTheme { w_ST_Theme }?,
  attribute w:eastAsiaTheme { w_ST_Theme }?,
  attribute w:cstheme { w_ST_Theme }?
w_EG_RPrBase =
  element rStyle { w_CT_String }?&
  element rFonts { w_CT_Fonts }?&
  element b { w_CT_OnOff }?&
  element bCs { w_CT_OnOff }?&
  element i { w_CT_OnOff }?&
  element iCs { w_CT_OnOff }?&
  element caps { w_CT_OnOff }?&
  element smallCaps { w_CT_OnOff }?&
  element strike { w_CT_OnOff }?&
  element dstrike { w_CT_OnOff }?&
  element outline { w_CT_OnOff }?&
  element shadow { w_CT_OnOff }?&
  element emboss { w_CT_OnOff }?&
  element imprint { w_CT_OnOff }?&
  element noProof { w_CT_OnOff }?&
  element snapToGrid { w_CT_OnOff }?&
  element vanish { w_CT_OnOff }?&
  element webHidden { w_CT_OnOff }?&
  element color { w_CT_Color }?&
  element spacing { w_CT_SignedTwipsMeasure }?&
  element w { w_CT_TextScale }?&
  element kern { w_CT_HpsMeasure }?&
  element position { w_CT_SignedHpsMeasure }?&
  element sz { w_CT_HpsMeasure }?&
  element szCs { w_CT_HpsMeasure }?&
  element highlight { w_CT_Highlight }?&
  element u { w_CT_Underline }?&
  element effect { w_CT_TextEffect }?&
  element bdr { w_CT_Border }?&
  element shd { w_CT_Shd }?&
  element fitText { w_CT_FitText }?&
  element vertAlign { w_CT_VerticalAlignRun }?&
  element rtl { w_CT_OnOff }?&
  element cs { w_CT_OnOff }?&
```

```
  element em { w_CT_Em }?&
  element lang { w_CT_Language }?&
  element eastAsianLayout { w_CT_EastAsianLayout }?&
  element specVanish { w_CT_OnOff }?&
  element oMath { w_CT_OnOff }?
w_EG_RPrContent =
  w_EG_RPrBase?,
  element rPrChange { w_CT_RPrChange }?
w_CT_RPr = w_EG_RPrContent?
w_EG_RPr = element rPr { w_CT_RPr }?
w_EG_RPrMath =
  w_EG_RPr
  | element ins { w_CT_MathCtrlIns }
  | element del { w_CT_MathCtrlDel }
w_CT_MathCtrlIns =
  w_CT_TrackChange,
  (element del { w_CT_RPrChange }
   | element rPr { w_CT_RPr })?
w_CT_MathCtrlDel =
  w_CT_TrackChange,
  (element rPr { w_CT_RPr })?
w_CT_RPrOriginal = w_EG_RPrBase*
w_CT_ParaRPrOriginal = w_EG_ParaRPrTrackChanges?, w_EG_RPrBase*
w_CT_ParaRPr =
  w_EG_ParaRPrTrackChanges?,
  w_EG_RPrBase?,
  element rPrChange { w_CT_ParaRPrChange }?
w_EG_ParaRPrTrackChanges =
  element ins { w_CT_TrackChange }?,
  element del { w_CT_TrackChange }?,
  element moveFrom { w_CT_TrackChange }?,
  element moveTo { w_CT_TrackChange }?
w_CT_AltChunk =
  r_id?,
  element altChunkPr { w_CT_AltChunkPr }?
w_CT_AltChunkPr = element matchSrc { w_CT_OnOff }?
w_ST_RubyAlign =
  string "center"
  | string "distributeLetter"
  | string "distributeSpace"
  | string "left"
  | string "right"
  | string "rightVertical"
w_CT_RubyAlign = attribute w:val { w_ST_RubyAlign }
w_CT_RubyPr =
  element rubyAlign { w_CT_RubyAlign },
  element hps { w_CT_HpsMeasure },
  element hpsRaise { w_CT_HpsMeasure },
  element hpsBaseText { w_CT_HpsMeasure },
  element lid { w_CT_Lang },
  element dirty { w_CT_OnOff }?
w_EG_RubyContent =
  element r { w_CT_R }
```



```
  | w_EG_RunLevelElts*
w_CT_RubyContent = w_EG_RubyContent*
w_CT_Ruby =
  element rubyPr { w_CT_RubyPr },
  element rt { w_CT_RubyContent },
  element rubyBase { w_CT_RubyContent }
w_ST_Lock =
  string "sdtLocked"
  | string "contentLocked"
  | string "unlocked"
  | string "sdtContentLocked"
w_CT_Lock = attribute w:val { w_ST_Lock }?
w_CT_SdtListItem =
  attribute w:displayText { s_ST_String }?,
  attribute w:value { s_ST_String }?
w_ST_SdtDateMappingType =
  string "text" | string "date" | string "dateTime"
w_CT_SdtDateMappingType = attribute w:val { w_ST_SdtDateMappingType }?
w_CT_CalendarType = attribute w:val { s_ST_CalendarType }?
w_CT_SdtDate =
  attribute w:fullDate { w_ST_DateTime }?,
  element dateFormat { w_CT_String }?,
  element lid { w_CT_Lang }?,
  element storeMappedDataAs { w_CT_SdtDateMappingType }?,
  element calendar { w_CT_CalendarType }?
w_CT_SdtComboBox =
  attribute w:lastValue { s_ST_String }?,
  element listItem { w_CT_SdtListItem }*
w_CT_SdtDocPart =
  element docPartGallery { w_CT_String }?,
  element docPartCategory { w_CT_String }?,
  element docPartUnique { w_CT_OnOff }?
w_CT_SdtDropDownList =
  attribute w:lastValue { s_ST_String }?,
  element listItem { w_CT_SdtListItem }*
w_CT_Placeholder = element docPart { w_CT_String }
w_CT_SdtText = attribute w:multiLine { s_ST_OnOff }?
w_CT_DataBinding =
  attribute w:prefixMappings { s_ST_String }?,
  attribute w:xpath { s_ST_String },
  attribute w:storeItemID { s_ST_String }
w_CT_SdtPr =
  element rPr { w_CT_RPr }?,
  element alias { w_CT_String }?,
  element tag { w_CT_String }?,
  element id { w_CT_DecimalNumber }?,
  element lock { w_CT_Lock }?,
  element placeholder { w_CT_Placeholder }?,
  element temporary { w_CT_OnOff }?,
  element showingPlcHdr { w_CT_OnOff }?,
  element dataBinding { w_CT_DataBinding }?,
  element label { w_CT_DecimalNumber }?,
  element tabIndex { w_CT_UnsignedDecimalNumber }?,
```

```
  (element equation { w_CT_Empty }
    | element comboBox { w_CT_SdtComboBox }
    | element date { w_CT_SdtDate }
    | element docPartObj { w_CT_SdtDocPart }
    | element docPartList { w_CT_SdtDocPart }
    | element dropDownList { w_CT_SdtDropDownList }
    | element picture { w_CT_Empty }
    | element richText { w_CT_Empty }
    | element text { w_CT_SdtText }
    | element citation { w_CT_Empty }
    | element group { w_CT_Empty }
    | element bibliography { w_CT_Empty })?
w_CT_SdtEndPr = (element rPr { w_CT_RPr }?)+
w_EG_ContentRunContent =
  element customXml { w_CT_CustomXmlRun }
  | element smartTag { w_CT_SmartTagRun }
  | element sdt { w_CT_SdtRun }
  | element dir { w_CT_DirContentRun }
  | element bdo { w_CT_BdoContentRun }
  | element r { w_CT_R }
  | w_EG_RunLevelElts*
w_CT_DirContentRun =
  attribute w:val { w_ST_Direction }?,
  w_EG_PContent*
w_CT_BdoContentRun =
  attribute w:val { w_ST_Direction }?,
  w_EG_PContent*
w_ST_Direction = string "ltr" | string "rtl"
w_CT_SdtContentRun = w_EG_PContent*
w_EG_ContentBlockContent =
  element customXml { w_CT_CustomXmlBlock }
  | element sdt { w_CT_SdtBlock }
  | element p { w_CT_P }*
  | element tbl { w_CT_Tbl }*
  | w_EG_RunLevelElts*
w_CT_SdtContentBlock = w_EG_ContentBlockContent*
w_EG_ContentRowContent =
  element tr { w_CT_Row }*
  | element customXml { w_CT_CustomXmlRow }
  | element sdt { w_CT_SdtRow }
  | w_EG_RunLevelElts*
w_CT_SdtContentRow = w_EG_ContentRowContent*
w_EG_ContentCellContent =
  element tc { w_CT_Tc }*
  | element customXml { w_CT_CustomXmlCell }
  | element sdt { w_CT_SdtCell }
  | w_EG_RunLevelElts*
w_CT_SdtContentCell = w_EG_ContentCellContent*
w_CT_SdtBlock =
  element sdtPr { w_CT_SdtPr }?,
  element sdtEndPr { w_CT_SdtEndPr }?,
  element sdtContent { w_CT_SdtContentBlock }?
w_CT_SdtRun =
```



```
  element sdtPr { w_CT_SdtPr }?,
  element sdtEndPr { w_CT_SdtEndPr }?,
  element sdtContent { w_CT_SdtContentRun }?
w_CT_SdtCell =
  element sdtPr { w_CT_SdtPr }?,
  element sdtEndPr { w_CT_SdtEndPr }?,
  element sdtContent { w_CT_SdtContentCell }?
w_CT_SdtRow =
  element sdtPr { w_CT_SdtPr }?,
  element sdtEndPr { w_CT_SdtEndPr }?,
  element sdtContent { w_CT_SdtContentRow }?
w_CT_Attr =
  attribute w:uri { s_ST_String }?,
  attribute w:name { s_ST_String },
  attribute w:val { s_ST_String }
w_CT_CustomXmlRun =
  attribute w:uri { s_ST_String }?,
  attribute w:element { s_ST_XmlName },
  element customXmlPr { w_CT_CustomXmlPr }?,
  w_EG_PContent*
w_CT_SmartTagRun =
  attribute w:uri { s_ST_String }?,
  attribute w:element { s_ST_XmlName },
  element smartTagPr { w_CT_SmartTagPr }?,
  w_EG_PContent*
w_CT_CustomXmlBlock =
  attribute w:uri { s_ST_String }?,
  attribute w:element { s_ST_XmlName },
  element customXmlPr { w_CT_CustomXmlPr }?,
  w_EG_ContentBlockContent*
w_CT_CustomXmlPr =
  element placeholder { w_CT_String }?,
  element attr { w_CT_Attr }*
w_CT_CustomXmlRow =
  attribute w:uri { s_ST_String }?,
  attribute w:element { s_ST_XmlName },
  element customXmlPr { w_CT_CustomXmlPr }?,
  w_EG_ContentRowContent*
w_CT_CustomXmlCell =
  attribute w:uri { s_ST_String }?,
  attribute w:element { s_ST_XmlName },
  element customXmlPr { w_CT_CustomXmlPr }?,
  w_EG_ContentCellContent*
w_CT_SmartTagPr = element attr { w_CT_Attr }*
w_EG_PContent =
  w_EG_ContentRunContent*
  | element fldSimple { w_CT_SimpleField }*
  | element hyperlink { w_CT_Hyperlink }
  | element subDoc { w_CT_Rel }
w_CT_P =
  attribute w:rsidRPr { w_ST_LongHexNumber }?,
  attribute w:rsidR { w_ST_LongHexNumber }?,
  attribute w:rsidDel { w_ST_LongHexNumber }?,
```

```
  attribute w:rsidP { w_ST_LongHexNumber }?,
  attribute w:rsidRDefault { w_ST_LongHexNumber }?,
  element pPr { w_CT_PPr }?,
  w_EG_PContent*
w_ST_TblWidth =
  string "nil" | string "pct" | string "dxa" | string "auto"
w_CT_Height =
  attribute w:val { s_ST_TwipsMeasure }?,
  attribute w:hRule { w_ST_HeightRule }?
w_ST_MeasurementOrPercent = w_ST_DecimalNumberOrPercent | s_ST_UniversalMeasure
w_CT_TblWidth =
  attribute w:w { w_ST_MeasurementOrPercent}?,
  attribute w:type { w_ST_TblWidth }?
w_CT_TblGridCol = attribute w:w { s_ST_TwipsMeasure }?
w_CT_TblGridBase = element gridCol { w_CT_TblGridCol }*
w_CT_TblGrid =
  w_CT_TblGridBase,
  element tblGridChange { w_CT_TblGridChange }?
w_CT_TcBorders =
  element top { w_CT_Border }?,
  element start { w_CT_Border }?,
  element left { w_CT_Border }?,
  element bottom { w_CT_Border }?,
  element end { w_CT_Border }?,
  element right { w_CT_Border }?,
  element insideH { w_CT_Border }?,
  element insideV { w_CT_Border }?,
  element tl2br { w_CT_Border }?,
  element tr2bl { w_CT_Border }?
w_CT_TcMar =
  element top { w_CT_TblWidth }?,
  element start { w_CT_TblWidth }?,
  element left { w_CT_TblWidth }?,
  element bottom { w_CT_TblWidth }?,
  element end { w_CT_TblWidth }?,
  element right { w_CT_TblWidth }?
w_ST_Merge = string "continue" | string "restart"
w_CT_VMerge = attribute w:val { w_ST_Merge }?
w_CT_HMerge = attribute w:val { w_ST_Merge }?
w_CT_TcPrBase =
  element cnfStyle { w_CT_Cnf }?,
  element tcW { w_CT_TblWidth }?,
  element gridSpan { w_CT_DecimalNumber }?,
  element hMerge { w_CT_HMerge }?,
  element vMerge { w_CT_VMerge }?,
  element tcBorders { w_CT_TcBorders }?,
  element shd { w_CT_Shd }?,
  element noWrap { w_CT_OnOff }?,
  element tcMar { w_CT_TcMar }?,
  element textDirection { w_CT_TextDirection }?,
  element tcFitText { w_CT_OnOff }?,
  element vAlign { w_CT_VerticalJc }?,
  element hideMark { w_CT_OnOff }?,
```



```
  element headers { w_CT_Headers }?
w_CT_TcPr =
  w_CT_TcPrInner,
  element tcPrChange { w_CT_TcPrChange }?
w_CT_TcPrInner = w_CT_TcPrBase, w_EG_CellMarkupElements?
w_CT_Tc =
  attribute w:id { s_ST_String }?,
  element tcPr { w_CT_TcPr }?,
  w_EG_BlockLevelElts+
w_ST_Cnf = xsd:string { length = "12" pattern = "[01]*" }
w_CT_Cnf =
  attribute w:val { w_ST_Cnf }?,
  attribute w:firstRow { s_ST_OnOff }?,
  attribute w:lastRow { s_ST_OnOff }?,
  attribute w:firstColumn { s_ST_OnOff }?,
  attribute w:lastColumn { s_ST_OnOff }?,
  attribute w:oddVBand { s_ST_OnOff }?,
  attribute w:evenVBand { s_ST_OnOff }?,
  attribute w:oddHBand { s_ST_OnOff }?,
  attribute w:evenHBand { s_ST_OnOff }?,
  attribute w:firstRowFirstColumn { s_ST_OnOff }?,
  attribute w:firstRowLastColumn { s_ST_OnOff }?,
  attribute w:lastRowFirstColumn { s_ST_OnOff }?,
  attribute w:lastRowLastColumn { s_ST_OnOff }?
w_CT_Headers = element header { w_CT_String }*
w_CT_TrPrBase =
  (element cnfStyle { w_CT_Cnf }?
   | element divId { w_CT_DecimalNumber }?
   | element gridBefore { w_CT_DecimalNumber }?
   | element gridAfter { w_CT_DecimalNumber }?
   | element wBefore { w_CT_TblWidth }?
   | element wAfter { w_CT_TblWidth }?
   | element cantSplit { w_CT_OnOff }?
   | element trHeight { w_CT_Height }?
   | element tblHeader { w_CT_OnOff }?
   | element tblCellSpacing { w_CT_TblWidth }?
   | element jc { w_CT_JcTable }?
   | element hidden { w_CT_OnOff }?)+
w_CT_TrPr =
  w_CT_TrPrBase,
  element ins { w_CT_TrackChange }?,
  element del { w_CT_TrackChange }?,
  element trPrChange { w_CT_TrPrChange }?
w_CT_Row =
  attribute w:rsidRPr { w_ST_LongHexNumber }?,
  attribute w:rsidR { w_ST_LongHexNumber }?,
  attribute w:rsidDel { w_ST_LongHexNumber }?,
  attribute w:rsidTr { w_ST_LongHexNumber }?,
  element tblPrEx { w_CT_TblPrEx }?,
  element trPr { w_CT_TrPr }?,
  w_EG_ContentCellContent*
w_ST_TblLayoutType = string "fixed" | string "autofit"
w_CT_TblLayoutType = attribute w:type { w_ST_TblLayoutType }?
```

```
w_ST_TblOverlap = string "never" | string "overlap"
w_CT_TblOverlap = attribute w:val { w_ST_TblOverlap }
w_CT_TblPPr =
  attribute w:leftFromText { s_ST_TwipsMeasure }?,
  attribute w:rightFromText { s_ST_TwipsMeasure }?,
  attribute w:topFromText { s_ST_TwipsMeasure }?,
  attribute w:bottomFromText { s_ST_TwipsMeasure }?,
  attribute w:vertAnchor { w_ST_VAnchor }?,
  attribute w:horzAnchor { w_ST_HAnchor }?,
  attribute w:tblpXSpec { s_ST_XAlign }?,
  attribute w:tblpX { w_ST_SignedTwipsMeasure }?,
  attribute w:tblpYSpec { s_ST_YAlign }?,
  attribute w:tblpY { w_ST_SignedTwipsMeasure }?
w_CT_TblCellMar =
  element top { w_CT_TblWidth }?,
  element start { w_CT_TblWidth }?,
  element left { w_CT_TblWidth }?,
  element bottom { w_CT_TblWidth }?,
  element end { w_CT_TblWidth }?,
  element right { w_CT_TblWidth }?
w_CT_TblBorders =
  element top { w_CT_Border }?,
  element start { w_CT_Border }?,
  element left { w_CT_Border }?,
  element bottom { w_CT_Border }?,
  element end { w_CT_Border }?,
  element right { w_CT_Border }?,
  element insideH { w_CT_Border }?,
  element insideV { w_CT_Border }?
w_CT_TblPrBase =
  element tblStyle { w_CT_String }?,
  element tblpPr { w_CT_TblPPr }?,
  element tblOverlap { w_CT_TblOverlap }?,
  element bidiVisual { w_CT_OnOff }?,
  element tblStyleRowBandSize { w_CT_DecimalNumber }?,
  element tblStyleColBandSize { w_CT_DecimalNumber }?,
  element tblW { w_CT_TblWidth }?,
  element jc { w_CT_JcTable }?,
  element tblCellSpacing { w_CT_TblWidth }?,
  element tblInd { w_CT_TblWidth }?,
  element tblBorders { w_CT_TblBorders }?,
  element shd { w_CT_Shd }?,
  element tblLayout { w_CT_TblLayoutType }?,
  element tblCellMar { w_CT_TblCellMar }?,
  element tblLook { w_CT_TblLook }?,
  element tblCaption { w_CT_String }?,
  element tblDescription { w_CT_String }?
w_CT_TblPr =
  w_CT_TblPrBase,
  element tblPrChange { w_CT_TblPrChange }?
w_CT_TblPrExBase =
  element tblW { w_CT_TblWidth }?,
  element jc { w_CT_JcTable }?,
```



```
  element tblCellSpacing { w_CT_TblWidth }?,
  element tblInd { w_CT_TblWidth }?,
  element tblBorders { w_CT_TblBorders }?,
  element shd { w_CT_Shd }?,
  element tblLayout { w_CT_TblLayoutType }?,
  element tblCellMar { w_CT_TblCellMar }?,
  element tblLook { w_CT_TblLook }?
w_CT_TblPrEx =
  w_CT_TblPrExBase,
  element tblPrExChange { w_CT_TblPrExChange }?
w_CT_Tbl =
  w_EG_RangeMarkupElements*,
  element tblPr { w_CT_TblPr },
  element tblGrid { w_CT_TblGrid },
  w_EG_ContentRowContent*
w_CT_TblLook =
  attribute w:firstRow { s_ST_OnOff }?,
  attribute w:lastRow { s_ST_OnOff }?,
  attribute w:firstColumn { s_ST_OnOff }?,
  attribute w:lastColumn { s_ST_OnOff }?,
  attribute w:noHBand { s_ST_OnOff }?,
  attribute w:noVBand { s_ST_OnOff }?,
  attribute w:val { w_ST_ShortHexNumber }?
w_ST_FtnPos =
  string "pageBottom"
  | string "beneathText"
  | string "sectEnd"
  | string "docEnd"
w_CT_FtnPos = attribute w:val { w_ST_FtnPos }
w_ST_EdnPos = string "sectEnd" | string "docEnd"
w_CT_EdnPos = attribute w:val { w_ST_EdnPos }
w_CT_NumFmt =
  attribute w:val { w_ST_NumberFormat },
  attribute w:format { s_ST_String }?
w_ST_RestartNumber =
  string "continuous" | string "eachSect" | string "eachPage"
w_CT_NumRestart = attribute w:val { w_ST_RestartNumber }
w_CT_FtnEdnRef =
  attribute w:customMarkFollows { s_ST_OnOff }?,
  attribute w:id { w_ST_DecimalNumber }
w_CT_FtnEdnSepRef = attribute w:id { w_ST_DecimalNumber }
w_CT_FtnEdn =
  attribute w:type { w_ST_FtnEdn }?,
  attribute w:id { w_ST_DecimalNumber },
  w_EG_BlockLevelElts+
w_EG_FtnEdnNumProps =
  element numStart { w_CT_DecimalNumber }?,
  element numRestart { w_CT_NumRestart }?
w_CT_FtnProps =
  element pos { w_CT_FtnPos }?,
  element numFmt { w_CT_NumFmt }?,
  w_EG_FtnEdnNumProps?
w_CT_EdnProps =
```

```
  element pos { w_CT_EdnPos }?,
  element numFmt { w_CT_NumFmt }?,
  w_EG_FtnEdnNumProps?
w_CT_FtnDocProps =
  w_CT_FtnProps,
  element footnote { w_CT_FtnEdnSepRef }*
w_CT_EdnDocProps =
  w_CT_EdnProps,
  element endnote { w_CT_FtnEdnSepRef }*
w_CT_RecipientData =
  element active { w_CT_OnOff }?,
  element column { w_CT_DecimalNumber },
  element uniqueTag { w_CT_Base64Binary}
w_CT_Base64Binary = attribute w:val { xsd:base64Binary }
w_CT_Recipients = element recipientData { w_CT_RecipientData }+
w_recipients = element recipients { w_CT_Recipients }
w_CT_OdsoFieldMapData =
  element type { w_CT_MailMergeOdsoFMDFieldType }?,
  element name { w_CT_String }?,
  element mappedName { w_CT_String }?,
  element column { w_CT_DecimalNumber }?,
  element lid { w_CT_Lang }?,
  element dynamicAddress { w_CT_OnOff }?
w_ST_MailMergeSourceType =
  string "database"
  | string "addressBook"
  | string "document1"
  | string "document2"
  | string "text"
  | string "email"
  | string "native"
  | string "legacy"
  | string "master"
w_CT_MailMergeSourceType = attribute w:val { w_ST_MailMergeSourceType }
w_CT_Odso =
  element udl { w_CT_String }?,
  element table { w_CT_String }?,
  element src { w_CT_Rel }?,
  element colDelim { w_CT_DecimalNumber }?,
  element type { w_CT_MailMergeSourceType }?,
  element fHdr { w_CT_OnOff }?,
  element fieldMapData { w_CT_OdsoFieldMapData }*,
  element recipientData { w_CT_Rel }*
w_CT_MailMerge =
  element mainDocumentType { w_CT_MailMergeDocType },
  element linkToQuery { w_CT_OnOff }?,
  element dataType { w_CT_MailMergeDataType },
  element connectString { w_CT_String }?,
  element query { w_CT_String }?,
  element dataSource { w_CT_Rel }?,
  element headerSource { w_CT_Rel }?,
  element doNotSuppressBlankLines { w_CT_OnOff }?,
  element destination { w_CT_MailMergeDest }?,
```



```
  element addressFieldName { w_CT_String }?,
  element mailSubject { w_CT_String }?,
  element mailAsAttachment { w_CT_OnOff }?,
  element viewMergedData { w_CT_OnOff }?,
  element activeRecord { w_CT_DecimalNumber }?,
  element checkErrors { w_CT_DecimalNumber }?,
  element odso { w_CT_Odso }?
w_ST_TargetScreenSz =
  string "544x376"
  | string "640x480"
  | string "720x512"
  | string "800x600"
  | string "1024x768"
  | string "1152x882"
  | string "1152x900"
  | string "1280x1024"
  | string "1600x1200"
  | string "1800x1440"
  | string "1920x1200"
w_CT_TargetScreenSz = attribute w:val { w_ST_TargetScreenSz }
w_CT_Compat =
  element useSingleBorderforContiguousCells { w_CT_OnOff }?,
  element wpJustification { w_CT_OnOff }?,
  element noTabHangInd { w_CT_OnOff }?,
  element noLeading { w_CT_OnOff }?,
  element spaceForUL { w_CT_OnOff }?,
  element noColumnBalance { w_CT_OnOff }?,
  element balanceSingleByteDoubleByteWidth { w_CT_OnOff }?,
  element noExtraLineSpacing { w_CT_OnOff }?,
  element doNotLeaveBackslashAlone { w_CT_OnOff }?,
  element ulTrailSpace { w_CT_OnOff }?,
  element doNotExpandShiftReturn { w_CT_OnOff }?,
  element spacingInWholePoints { w_CT_OnOff }?,
  element lineWrapLikeWord6 { w_CT_OnOff }?,
  element printBodyTextBeforeHeader { w_CT_OnOff }?,
  element printColBlack { w_CT_OnOff }?,
  element wpSpaceWidth { w_CT_OnOff }?,
  element showBreaksInFrames { w_CT_OnOff }?,
  element subFontBySize { w_CT_OnOff }?,
  element suppressBottomSpacing { w_CT_OnOff }?,
  element suppressTopSpacing { w_CT_OnOff }?,
  element suppressSpacingAtTopOfPage { w_CT_OnOff }?,
  element suppressTopSpacingWP { w_CT_OnOff }?,
  element suppressSpBfAfterPgBrk { w_CT_OnOff }?,
  element swapBordersFacingPages { w_CT_OnOff }?,
  element convMailMergeEsc { w_CT_OnOff }?,
  element truncateFontHeightsLikeWP6 { w_CT_OnOff }?,
  element mwSmallCaps { w_CT_OnOff }?,
  element usePrinterMetrics { w_CT_OnOff }?,
  element doNotSuppressParagraphBorders { w_CT_OnOff }?,
  element wrapTrailSpaces { w_CT_OnOff }?,
  element footnoteLayoutLikeWW8 { w_CT_OnOff }?,
  element shapeLayoutLikeWW8 { w_CT_OnOff }?,
```

```
  element alignTablesRowByRow { w_CT_OnOff }?,
  element forgetLastTabAlignment { w_CT_OnOff }?,
  element adjustLineHeightInTable { w_CT_OnOff }?,
  element autoSpaceLikeWord95 { w_CT_OnOff }?,
  element noSpaceRaiseLower { w_CT_OnOff }?,
  element doNotUseHTMLParagraphAutoSpacing { w_CT_OnOff }?,
  element layoutRawTableWidth { w_CT_OnOff }?,
  element layoutTableRowsApart { w_CT_OnOff }?,
  element useWord97LineBreakRules { w_CT_OnOff }?,
  element doNotBreakWrappedTables { w_CT_OnOff }?,
  element doNotSnapToGridInCell { w_CT_OnOff }?,
  element selectFldWithFirstOrLastChar { w_CT_OnOff }?,
  element applyBreakingRules { w_CT_OnOff }?,
  element doNotWrapTextWithPunct { w_CT_OnOff }?,
  element doNotUseEastAsianBreakRules { w_CT_OnOff }?,
  element useWord2002TableStyleRules { w_CT_OnOff }?,
  element growAutofit { w_CT_OnOff }?,
  element useFELayout { w_CT_OnOff }?,
  element useNormalStyleForList { w_CT_OnOff }?,
  element doNotUseIndentAsNumberingTabStop { w_CT_OnOff }?,
  element useAltKinsokuLineBreakRules { w_CT_OnOff }?,
  element allowSpaceOfSameStyleInTable { w_CT_OnOff }?,
  element doNotSuppressIndentation { w_CT_OnOff }?,
  element doNotAutofitConstrainedTables { w_CT_OnOff }?,
  element autofitToFirstFixedWidthCell { w_CT_OnOff }?,
  element underlineTabInNumList { w_CT_OnOff }?,
  element displayHangulFixedWidth { w_CT_OnOff }?,
  element splitPgBreakAndParaMark { w_CT_OnOff }?,
  element doNotVertAlignCellWithSp { w_CT_OnOff }?,
  element doNotBreakConstrainedForcedTable { w_CT_OnOff }?,
  element doNotVertAlignInTxbx { w_CT_OnOff }?,
  element useAnsiKerningPairs { w_CT_OnOff }?,
  element cachedColBalance { w_CT_OnOff }?,
  element compatSetting { w_CT_CompatSetting }*
w_CT_CompatSetting =
  attribute w:name { s_ST_String }?,
  attribute w:uri { s_ST_String }?,
  attribute w:val { s_ST_String }?
w_CT_DocVar =
  attribute w:name { s_ST_String },
  attribute w:val { s_ST_String }
w_CT_DocVars = element docVar { w_CT_DocVar }*
w_CT_DocRsids =
  element rsidRoot { w_CT_LongHexNumber }?,
  element rsid { w_CT_LongHexNumber }*
w_ST_CharacterSpacing =
  string "doNotCompress"
  | string "compressPunctuation"
  | string "compressPunctuationAndJapaneseKana"
w_CT_CharacterSpacing = attribute w:val { w_ST_CharacterSpacing }
w_CT_SaveThroughXslt =
  r_id?,
  attribute w:solutionID { s_ST_String }?
```



```
w_CT_RPrDefault = element rPr { w_CT_RPr }?
w_CT_PPrDefault = element pPr { w_CT_PPrGeneral }?
w_CT_DocDefaults =
  element rPrDefault { w_CT_RPrDefault }?,
  element pPrDefault { w_CT_PPrDefault }?
w_ST_WmlColorSchemeIndex =
  string "dark1"
  | string "light1"
  | string "dark2"
  | string "light2"
  | string "accent1"
  | string "accent2"
  | string "accent3"
  | string "accent4"
  | string "accent5"
  | string "accent6"
  | string "hyperlink"
  | string "followedHyperlink"
w_CT_ColorSchemeMapping =
  attribute w:bg1 { w_ST_WmlColorSchemeIndex }?,
  attribute w:t1 { w_ST_WmlColorSchemeIndex }?,
  attribute w:bg2 { w_ST_WmlColorSchemeIndex }?,
  attribute w:t2 { w_ST_WmlColorSchemeIndex }?,
  attribute w:accent1 { w_ST_WmlColorSchemeIndex }?,
  attribute w:accent2 { w_ST_WmlColorSchemeIndex }?,
  attribute w:accent3 { w_ST_WmlColorSchemeIndex }?,
  attribute w:accent4 { w_ST_WmlColorSchemeIndex }?,
  attribute w:accent5 { w_ST_WmlColorSchemeIndex }?,
  attribute w:accent6 { w_ST_WmlColorSchemeIndex }?,
  attribute w:hyperlink { w_ST_WmlColorSchemeIndex }?,
  attribute w:followedHyperlink { w_ST_WmlColorSchemeIndex }?
w_CT_ReadingModeInkLockDown =
  attribute w:actualPg { s_ST_OnOff },
  attribute w:w { w_ST_PixelsMeasure },
  attribute w:h { w_ST_PixelsMeasure },
  attribute w:fontSz { w_ST_DecimalNumberOrPercent }
w_CT_WriteProtection =
  attribute w:recommended { s_ST_OnOff }?,
  w_AG_Password,
  w_AG_TransitionalPassword
w_CT_Settings =
  element writeProtection { w_CT_WriteProtection }?,
  element view { w_CT_View }?,
  element zoom { w_CT_Zoom }?,
  element removePersonalInformation { w_CT_OnOff }?,
  element removeDateAndTime { w_CT_OnOff }?,
  element doNotDisplayPageBoundaries { w_CT_OnOff }?,
  element displayBackgroundShape { w_CT_OnOff }?,
  element printPostScriptOverText { w_CT_OnOff }?,
  element printFractionalCharacterWidth { w_CT_OnOff }?,
  element printFormsData { w_CT_OnOff }?,
  element embedTrueTypeFonts { w_CT_OnOff }?,
  element embedSystemFonts { w_CT_OnOff }?,
```

```
  element saveSubsetFonts { w_CT_OnOff }?,
  element saveFormsData { w_CT_OnOff }?,
  element mirrorMargins { w_CT_OnOff }?,
  element alignBordersAndEdges { w_CT_OnOff }?,
  element bordersDoNotSurroundHeader { w_CT_OnOff }?,
  element bordersDoNotSurroundFooter { w_CT_OnOff }?,
  element gutterAtTop { w_CT_OnOff }?,
  element hideSpellingErrors { w_CT_OnOff }?,
  element hideGrammaticalErrors { w_CT_OnOff }?,
  element activeWritingStyle { w_CT_WritingStyle }*,
  element proofState { w_CT_Proof }?,
  element formsDesign { w_CT_OnOff }?,
  element attachedTemplate { w_CT_Rel }?,
  element linkStyles { w_CT_OnOff }?,
  element stylePaneFormatFilter { w_CT_StylePaneFilter }?,
  element stylePaneSortMethod { w_CT_StyleSort }?,
  element documentType { w_CT_DocType }?,
  element mailMerge { w_CT_MailMerge }?,
  element revisionView { w_CT_TrackChangesView }?,
  element trackRevisions { w_CT_OnOff }?,
  element doNotTrackMoves { w_CT_OnOff }?,
  element doNotTrackFormatting { w_CT_OnOff }?,
  element documentProtection { w_CT_DocProtect }?,
  element autoFormatOverride { w_CT_OnOff }?,
  element styleLockTheme { w_CT_OnOff }?,
  element styleLockQFSet { w_CT_OnOff }?,
  element defaultTabStop { w_CT_TwipsMeasure }?,
  element autoHyphenation { w_CT_OnOff }?,
  element consecutiveHyphenLimit { w_CT_DecimalNumber }?,
  element hyphenationZone { w_CT_TwipsMeasure }?,
  element doNotHyphenateCaps { w_CT_OnOff }?,
  element showEnvelope { w_CT_OnOff }?,
  element summaryLength { w_CT_DecimalNumberOrPrecent }?,
  element clickAndTypeStyle { w_CT_String }?,
  element defaultTableStyle { w_CT_String }?,
  element evenAndOddHeaders { w_CT_OnOff }?,
  element bookFoldRevPrinting { w_CT_OnOff }?,
  element bookFoldPrinting { w_CT_OnOff }?,
  element bookFoldPrintingSheets { w_CT_DecimalNumber }?,
  element drawingGridHorizontalSpacing { w_CT_TwipsMeasure }?,
  element drawingGridVerticalSpacing { w_CT_TwipsMeasure }?,
  element displayHorizontalDrawingGridEvery { w_CT_DecimalNumber }?,
  element displayVerticalDrawingGridEvery { w_CT_DecimalNumber }?,
  element doNotUseMarginsForDrawingGridOrigin { w_CT_OnOff }?,
  element drawingGridHorizontalOrigin { w_CT_TwipsMeasure }?,
  element drawingGridVerticalOrigin { w_CT_TwipsMeasure }?,
  element doNotShadeFormData { w_CT_OnOff }?,
  element noPunctuationKerning { w_CT_OnOff }?,
  element characterSpacingControl { w_CT_CharacterSpacing }?,
  element printTwoOnOne { w_CT_OnOff }?,
  element strictFirstAndLastChars { w_CT_OnOff }?,
  element noLineBreaksAfter { w_CT_Kinsoku }?,
  element noLineBreaksBefore { w_CT_Kinsoku }?,
```



```
  element savePreviewPicture { w_CT_OnOff }?,
  element doNotValidateAgainstSchema { w_CT_OnOff }?,
  element saveInvalidXml { w_CT_OnOff }?,
  element ignoreMixedContent { w_CT_OnOff }?,
  element alwaysShowPlaceholderText { w_CT_OnOff }?,
  element doNotDemarcateInvalidXml { w_CT_OnOff }?,
  element saveXmlDataOnly { w_CT_OnOff }?,
  element useXSLTWhenSaving { w_CT_OnOff }?,
  element saveThroughXslt { w_CT_SaveThroughXslt }?,
  element showXMLTags { w_CT_OnOff }?,
  element alwaysMergeEmptyNamespace { w_CT_OnOff }?,
  element updateFields { w_CT_OnOff }?,
  element hdrShapeDefaults { w_CT_ShapeDefaults }?,
  element footnotePr { w_CT_FtnDocProps }?,
  element endnotePr { w_CT_EdnDocProps }?,
  element compat { w_CT_Compat }?,
  element docVars { w_CT_DocVars }?,
  element rsids { w_CT_DocRsids }?,
  m_mathPr?,
  element attachedSchema { w_CT_String }*,
  element themeFontLang { w_CT_Language }?,
  element clrSchemeMapping { w_CT_ColorSchemeMapping }?,
  element doNotIncludeSubdocsInStats { w_CT_OnOff }?,
  element doNotAutoCompressPictures { w_CT_OnOff }?,
  element forceUpgrade { w_CT_Empty }?,
  element captions { w_CT_Captions }?,
  element readModeInkLockDown { w_CT_ReadingModeInkLockDown }?,
  element smartTagType { w_CT_SmartTagType }*,
  sl_schemaLibrary?,
  element shapeDefaults { w_CT_ShapeDefaults }?,
  element doNotEmbedSmartTags { w_CT_OnOff }?,
  element decimalSymbol { w_CT_String }?,
  element listSeparator { w_CT_String }?
w_CT_StyleSort = attribute w:val { w_ST_StyleSort }
w_CT_StylePaneFilter =
  attribute w:allStyles { s_ST_OnOff }?,
  attribute w:customStyles { s_ST_OnOff }?,
  attribute w:latentStyles { s_ST_OnOff }?,
  attribute w:stylesInUse { s_ST_OnOff }?,
  attribute w:headingStyles { s_ST_OnOff }?,
  attribute w:numberingStyles { s_ST_OnOff }?,
  attribute w:tableStyles { s_ST_OnOff }?,
  attribute w:directFormattingOnRuns { s_ST_OnOff }?,
  attribute w:directFormattingOnParagraphs { s_ST_OnOff }?,
  attribute w:directFormattingOnNumbering { s_ST_OnOff }?,
  attribute w:directFormattingOnTables { s_ST_OnOff }?,
  attribute w:clearFormatting { s_ST_OnOff }?,
  attribute w:top3HeadingStyles { s_ST_OnOff }?,
  attribute w:visibleStyles { s_ST_OnOff }?,
  attribute w:alternateStyleNames { s_ST_OnOff }?,
  attribute w:val { w_ST_ShortHexNumber }?
w_ST_StyleSort =
  string "name"
```

```
  | string "priority"
  | string "default"
  | string "font"
  | string "basedOn"
  | string "type"
  | string "0000"
  | string "0001"
  | string "0002"
  | string "0003"
  | string "0004"
  | string "0005"
w_CT_WebSettings =
  element frameset { w_CT_Frameset }?,
  element divs { w_CT_Divs }?,
  element encoding { w_CT_String }?,
  element optimizeForBrowser { w_CT_OptimizeForBrowser }?,
  element relyOnVML { w_CT_OnOff }?,
  element allowPNG { w_CT_OnOff }?,
  element doNotRelyOnCSS { w_CT_OnOff }?,
  element doNotSaveAsSingleFile { w_CT_OnOff }?,
  element doNotOrganizeInFolder { w_CT_OnOff }?,
  element doNotUseLongFileNames { w_CT_OnOff }?,
  element pixelsPerInch { w_CT_DecimalNumber }?,
  element targetScreenSz { w_CT_TargetScreenSz }?,
  element saveSmartTagsAsXml { w_CT_OnOff }?
w_ST_FrameScrollbar = string "on" | string "off" | string "auto"
w_CT_FrameScrollbar = attribute w:val { w_ST_FrameScrollbar }
w_CT_OptimizeForBrowser =
  w_CT_OnOff,
  attribute w:target { s_ST_String }?
w_CT_Frame =
  element sz { w_CT_String }?,
  element name { w_CT_String }?,
  element title { w_CT_String }?,
  element longDesc { w_CT_Rel }?,
  element sourceFileName { w_CT_Rel }?,
  element marW { w_CT_PixelsMeasure }?,
  element marH { w_CT_PixelsMeasure }?,
  element scrollbar { w_CT_FrameScrollbar }?,
  element noResizeAllowed { w_CT_OnOff }?,
  element linkedToFile { w_CT_OnOff }?
w_ST_FrameLayout = string "rows" | string "cols" | string "none"
w_CT_FrameLayout = attribute w:val { w_ST_FrameLayout }
w_CT_FramesetSplitbar =
  element w { w_CT_TwipsMeasure }?,
  element color { w_CT_Color }?,
  element noBorder { w_CT_OnOff }?,
  element flatBorders { w_CT_OnOff }?
w_CT_Frameset =
  element sz { w_CT_String }?,
  element framesetSplitbar { w_CT_FramesetSplitbar }?,
  element frameLayout { w_CT_FrameLayout }?,
  element title { w_CT_String }?,
```



```
  (element frameset { w_CT_Frameset }*
   | element frame { w_CT_Frame }*)*
w_CT_NumPicBullet =
  attribute w:numPicBulletId { w_ST_DecimalNumber },
  (element pict { w_CT_Picture }
   | element drawing { w_CT_Drawing })
w_ST_LevelSuffix = string "tab" | string "space" | string "nothing"
w_CT_LevelSuffix = attribute w:val { w_ST_LevelSuffix }
w_CT_LevelText =
  attribute w:val { s_ST_String }?,
  attribute w:null { s_ST_OnOff }?
w_CT_LvlLegacy =
  attribute w:legacy { s_ST_OnOff }?,
  attribute w:legacySpace { s_ST_TwipsMeasure }?,
  attribute w:legacyIndent { w_ST_SignedTwipsMeasure }?
w_CT_Lvl =
  attribute w:ilvl { w_ST_DecimalNumber },
  attribute w:tplc { w_ST_LongHexNumber }?,
  attribute w:tentative { s_ST_OnOff }?,
  element start { w_CT_DecimalNumber }?,
  element numFmt { w_CT_NumFmt }?,
  element lvlRestart { w_CT_DecimalNumber }?,
  element pStyle { w_CT_String }?,
  element isLgl { w_CT_OnOff }?,
  element suff { w_CT_LevelSuffix }?,
  element lvlText { w_CT_LevelText }?,
  element lvlPicBulletId { w_CT_DecimalNumber }?,
  element legacy { w_CT_LvlLegacy }?,
  element lvlJc { w_CT_Jc }?,
  element pPr { w_CT_PPrGeneral }?,
  element rPr { w_CT_RPr }?
w_ST_MultiLevelType =
  string "singleLevel" | string "multilevel" | string "hybridMultilevel"
w_CT_MultiLevelType = attribute w:val { w_ST_MultiLevelType }
w_CT_AbstractNum =
  attribute w:abstractNumId { w_ST_DecimalNumber },
  element nsid { w_CT_LongHexNumber }?,
  element multiLevelType { w_CT_MultiLevelType }?,
  element tmpl { w_CT_LongHexNumber }?,
  element name { w_CT_String }?,
  element styleLink { w_CT_String }?,
  element numStyleLink { w_CT_String }?,
  element lvl { w_CT_Lvl }*
w_CT_NumLvl =
  attribute w:ilvl { w_ST_DecimalNumber },
  element startOverride { w_CT_DecimalNumber }?,
  element lvl { w_CT_Lvl }?
w_CT_Num =
  attribute w:numId { w_ST_DecimalNumber },
  element abstractNumId { w_CT_DecimalNumber },
  element lvlOverride { w_CT_NumLvl }*
w_CT_Numbering =
  element numPicBullet { w_CT_NumPicBullet }*,
```

```
  element abstractNum { w_CT_AbstractNum }*,
  element num { w_CT_Num }*,
  element numIdMacAtCleanup { w_CT_DecimalNumber }?
w_ST_TblStyleOverrideType =
  string "wholeTable"
  | string "firstRow"
  | string "lastRow"
  | string "firstCol"
  | string "lastCol"
  | string "band1Vert"
  | string "band2Vert"
  | string "band1Horz"
  | string "band2Horz"
  | string "neCell"
  | string "nwCell"
  | string "seCell"
  | string "swCell"
w_CT_TblStylePr =
  attribute w:type { w_ST_TblStyleOverrideType },
  element pPr { w_CT_PPrGeneral }?,
  element rPr { w_CT_RPr }?,
  element tblPr { w_CT_TblPrBase }?,
  element trPr { w_CT_TrPr }?,
  element tcPr { w_CT_TcPr }?
w_ST_StyleType =
  string "paragraph"
  | string "character"
  | string "table"
  | string "numbering"
w_CT_Style =
  attribute w:type { w_ST_StyleType }?,
  attribute w:styleId { s_ST_String }?,
  attribute w:default { s_ST_OnOff }?,
  attribute w:customStyle { s_ST_OnOff }?,
  element name { w_CT_String }?,
  element aliases { w_CT_String }?,
  element basedOn { w_CT_String }?,
  element next { w_CT_String }?,
  element link { w_CT_String }?,
  element autoRedefine { w_CT_OnOff }?,
  element hidden { w_CT_OnOff }?,
  element uiPriority { w_CT_DecimalNumber }?,
  element semiHidden { w_CT_OnOff }?,
  element unhideWhenUsed { w_CT_OnOff }?,
  element qFormat { w_CT_OnOff }?,
  element locked { w_CT_OnOff }?,
  element personal { w_CT_OnOff }?,
  element personalCompose { w_CT_OnOff }?,
  element personalReply { w_CT_OnOff }?,
  element rsid { w_CT_LongHexNumber }?,
  element pPr { w_CT_PPrGeneral }?,
  element rPr { w_CT_RPr }?,
  element tblPr { w_CT_TblPrBase }?,
```



```
  element trPr { w_CT_TrPr }?,
  element tcPr { w_CT_TcPr }?,
  element tblStylePr { w_CT_TblStylePr }*
w_CT_LsdException =
  attribute w:name { s_ST_String },
  attribute w:locked { s_ST_OnOff }?,
  attribute w:uiPriority { w_ST_DecimalNumber }?,
  attribute w:semiHidden { s_ST_OnOff }?,
  attribute w:unhideWhenUsed { s_ST_OnOff }?,
  attribute w:qFormat { s_ST_OnOff }?
w_CT_LatentStyles =
  attribute w:defLockedState { s_ST_OnOff }?,
  attribute w:defUIPriority { w_ST_DecimalNumber }?,
  attribute w:defSemiHidden { s_ST_OnOff }?,
  attribute w:defUnhideWhenUsed { s_ST_OnOff }?,
  attribute w:defQFormat { s_ST_OnOff }?,
  attribute w:count { w_ST_DecimalNumber }?,
  element lsdException { w_CT_LsdException }*
w_CT_Styles =
  element docDefaults { w_CT_DocDefaults }?,
  element latentStyles { w_CT_LatentStyles }?,
  element style { w_CT_Style }*
w_CT_Panose = attribute w:val { s_ST_Panose }
w_ST_FontFamily =
  string "decorative"
  | string "modern"
  | string "roman"
  | string "script"
  | string "swiss"
  | string "auto"
w_CT_FontFamily = attribute w:val { w_ST_FontFamily }
w_ST_Pitch = string "fixed" | string "variable" | string "default"
w_CT_Pitch = attribute w:val { w_ST_Pitch }
w_CT_FontSig =
  attribute w:usb0 { w_ST_LongHexNumber },
  attribute w:usb1 { w_ST_LongHexNumber },
  attribute w:usb2 { w_ST_LongHexNumber },
  attribute w:usb3 { w_ST_LongHexNumber },
  attribute w:csb0 { w_ST_LongHexNumber },
  attribute w:csb1 { w_ST_LongHexNumber }
w_CT_FontRel =
  w_CT_Rel,
  attribute w:fontKey { s_ST_Guid }?,
  attribute w:subsetted { s_ST_OnOff }?
w_CT_Font =
  attribute w:name { s_ST_String },
  element altName { w_CT_String }?,
  element panose1 { w_CT_Panose }?,
  element charset { w_CT_Charset }?,
  element family { w_CT_FontFamily }?,
  element notTrueType { w_CT_OnOff }?,
  element pitch { w_CT_Pitch }?,
  element sig { w_CT_FontSig }?,
```

```
  element embedRegular { w_CT_FontRel }?,
  element embedBold { w_CT_FontRel }?,
  element embedItalic { w_CT_FontRel }?,
  element embedBoldItalic { w_CT_FontRel }?
w_CT_FontsList = element font { w_CT_Font }*
w_CT_DivBdr =
  element top { w_CT_Border }?,
  element left { w_CT_Border }?,
  element bottom { w_CT_Border }?,
  element right { w_CT_Border }?
w_CT_Div =
  attribute w:id { w_ST_DecimalNumber },
  element blockQuote { w_CT_OnOff }?,
  element bodyDiv { w_CT_OnOff }?,
  element marLeft { w_CT_SignedTwipsMeasure },
  element marRight { w_CT_SignedTwipsMeasure },
  element marTop { w_CT_SignedTwipsMeasure },
  element marBottom { w_CT_SignedTwipsMeasure },
  element divBdr { w_CT_DivBdr }?,
  element divsChild { w_CT_Divs }*
w_CT_Divs = element div { w_CT_Div }+
w_CT_TxbxContent = w_EG_BlockLevelElts+
w_txbxContent = element txbxContent { w_CT_TxbxContent }
w_EG_MathContent = m_oMathPara | m_oMath
w_EG_BlockLevelChunkElts = w_EG_ContentBlockContent*
w_EG_BlockLevelElts =
  w_EG_BlockLevelChunkElts*
  | element altChunk { w_CT_AltChunk }*
w_EG_RunLevelElts =
  element proofErr { w_CT_ProofErr }?
  | element permStart { w_CT_PermStart }?
  | element permEnd { w_CT_Perm }?
  | w_EG_RangeMarkupElements*
  | element ins { w_CT_RunTrackChange }?
  | element del { w_CT_RunTrackChange }?
  | element moveFrom { w_CT_RunTrackChange }
  | element moveTo { w_CT_RunTrackChange }
  | w_EG_MathContent*
w_CT_Body =
  w_EG_BlockLevelElts*,
  element sectPr { w_CT_SectPr }?
w_CT_ShapeDefaults = (w_any_vml_office*)+
w_CT_Comments = element comment { w_CT_Comment }*
w_comments = element comments { w_CT_Comments }
w_CT_Footnotes = element footnote { w_CT_FtnEdn }*
w_footnotes = element footnotes { w_CT_Footnotes }
w_CT_Endnotes = element endnote { w_CT_FtnEdn }*
w_endnotes = element endnotes { w_CT_Endnotes }
w_hdr = element hdr { w_CT_HdrFtr }
w_ftr = element ftr { w_CT_HdrFtr }
w_CT_SmartTagType =
  attribute w:namespaceuri { s_ST_String }?,
  attribute w:name { s_ST_String }?,
```



```
  attribute w:url { s_ST_String }?
w_ST_ThemeColor =
  string "dark1"
  | string "light1"
  | string "dark2"
  | string "light2"
  | string "accent1"
  | string "accent2"
  | string "accent3"
  | string "accent4"
  | string "accent5"
  | string "accent6"
  | string "hyperlink"
  | string "followedHyperlink"
  | string "none"
  | string "background1"
  | string "text1"
  | string "background2"
  | string "text2"
w_ST_DocPartBehavior = string "content" | string "p" | string "pg"
w_CT_DocPartBehavior = attribute w:val { w_ST_DocPartBehavior }
w_CT_DocPartBehaviors = element behavior { w_CT_DocPartBehavior }+
w_ST_DocPartType =
  string "none"
  | string "normal"
  | string "autoExp"
  | string "toolbar"
  | string "speller"
  | string "formFld"
  | string "bbPlcHdr"
w_CT_DocPartType = attribute w:val { w_ST_DocPartType }
w_CT_DocPartTypes =
  attribute w:all { s_ST_OnOff }?,
  (element type { w_CT_DocPartType }+)
w_ST_DocPartGallery =
  string "placeholder"
  | string "any"
  | string "default"
  | string "docParts"
  | string "coverPg"
  | string "eq"
  | string "ftrs"
  | string "hdrs"
  | string "pgNum"
  | string "tbls"
  | string "watermarks"
  | string "autoTxt"
  | string "txtBox"
  | string "pgNumT"
  | string "pgNumB"
  | string "pgNumMargins"
  | string "tblOfContents"
  | string "bib"
```

```
  | string "custQuickParts"
  | string "custCoverPg"
  | string "custEq"
  | string "custFtrs"
  | string "custHdrs"
  | string "custPgNum"
  | string "custTbls"
  | string "custWatermarks"
  | string "custAutoTxt"
  | string "custTxtBox"
  | string "custPgNumT"
  | string "custPgNumB"
  | string "custPgNumMargins"
  | string "custTblOfContents"
  | string "custBib"
  | string "custom1"
  | string "custom2"
  | string "custom3"
  | string "custom4"
  | string "custom5"
w_CT_DocPartGallery = attribute w:val { w_ST_DocPartGallery }
w_CT_DocPartCategory =
  element name { w_CT_String },
  element gallery { w_CT_DocPartGallery }
w_CT_DocPartName =
  attribute w:val { s_ST_String },
  attribute w:decorated { s_ST_OnOff }?
w_CT_DocPartPr =
  element name { w_CT_DocPartName }&
  element style { w_CT_String }?&
  element category { w_CT_DocPartCategory }?&
  element types { w_CT_DocPartTypes }?&
  element behaviors { w_CT_DocPartBehaviors }?&
  element description { w_CT_String }?&
  element guid { w_CT_Guid }?
w_CT_DocPart =
  element docPartPr { w_CT_DocPartPr }?,
  element docPartBody { w_CT_Body }?
w_CT_DocParts = element docPart { w_CT_DocPart }+
w_settings = element settings { w_CT_Settings }
w_webSettings = element webSettings { w_CT_WebSettings }
w_fonts = element fonts { w_CT_FontsList }
w_numbering = element numbering { w_CT_Numbering }
w_styles = element styles { w_CT_Styles }
w_ST_CaptionPos =
  string "above" | string "below" | string "left" | string "right"
w_CT_Caption =
  attribute w:name { s_ST_String },
  attribute w:pos { w_ST_CaptionPos }?,
  attribute w:chapNum { s_ST_OnOff }?,
  attribute w:heading { w_ST_DecimalNumber }?,
  attribute w:noLabel { s_ST_OnOff }?,
  attribute w:numFmt { w_ST_NumberFormat }?,
```



```
  attribute w:sep { w_ST_ChapterSep }?
w_CT_AutoCaption =
  attribute w:name { s_ST_String },
  attribute w:caption { s_ST_String }
w_CT_AutoCaptions = element autoCaption { w_CT_AutoCaption }+
w_CT_Captions =
  element caption { w_CT_Caption }+,
  element autoCaptions { w_CT_AutoCaptions }?
w_CT_DocumentBase = element background { w_CT_Background }?
w_CT_Document =
  w_CT_DocumentBase,
  element body { w_CT_Body }?,
  attribute w:conformance { s_ST_ConformanceClass }?
w_CT_GlossaryDocument =
  w_CT_DocumentBase,
  element docParts { w_CT_DocParts }?
w_document = element document { w_CT_Document }
w_glossaryDocument = element glossaryDocument { w_CT_GlossaryDocument }
w_any_vml_office =
  o_shapedefaults
  | o_shapelayout
  | o_signatureline
  | o_ink
  | o_diagram
  | o_skew
  | o_extrusion
  | o_callout
  | o_lock
  | o_OLEObject
  | o_complex
  | o_left
  | o_top
  | o_right
  | o_bottom
  | o_column
  | o_clippath
  | o_fill
w_any_vml_vml =
  v_shape
  | v_shapetype
  | v_group
  | v_background
  | v_fill
  | v_formulas
  | v_handles
  | v_imagedata
  | v_path
  | v_textbox
  | v_shadow
  | v_stroke
  | v_textpath
  | v_arc
  | v_curve
```

```
  | v_image
  | v_line
  | v_oval
  | v_polyline
  | v_rect
  | v_roundrect
```

B.1.1 Part Schemas

B.1.1.1 Comments Part

This schema is available in the file WordprocessingML_Comments.rnc.

```
include "wml.rnc"
include "shared-relationshipReference.rnc"
include "dml-wordprocessingDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
include "vml-presentationDrawing.rnc"
include "xml.rnc"
include "shared-customXmlSchemaProperties.rnc"
include "vml-officeDrawing.rnc"
include "vml-main.rnc"
include "vml-spreadsheetDrawing.rnc"
include "vml-wordprocessingDrawing.rnc"
include "shared-math.rnc"
start = w_comments
```

B.1.1.2 Document Settings Part

This schema is available in the file WordprocessingML_Document_Settings.rnc.

```
include "wml.rnc"
include "shared-relationshipReference.rnc"
include "dml-wordprocessingDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
include "vml-presentationDrawing.rnc"
include "xml.rnc"
include "shared-customXmlSchemaProperties.rnc"
include "vml-officeDrawing.rnc"
```



```
include "vml-main.rnc"
include "vml-spreadsheetDrawing.rnc"
include "vml-wordprocessingDrawing.rnc"
include "shared-math.rnc"
start = w_settings
```

B.1.1.3 Endnotes Part

This schema is available in the file WordprocessingML_Endnotes.rnc.

```
include "wml.rnc"
include "shared-relationshipReference.rnc"
include "dml-wordprocessingDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
include "vml-presentationDrawing.rnc"
include "xml.rnc"
include "shared-customXmlSchemaProperties.rnc"
include "vml-officeDrawing.rnc"
include "vml-main.rnc"
include "vml-spreadsheetDrawing.rnc"
include "vml-wordprocessingDrawing.rnc"
include "shared-math.rnc"
start = w_endnotes
```

B.1.1.4 Font Table Part

This schema is available in the file WordprocessingML_Font_Table.rnc.

```
include "wml.rnc"
include "shared-relationshipReference.rnc"
include "dml-wordprocessingDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
include "vml-presentationDrawing.rnc"
include "xml.rnc"
include "shared-customXmlSchemaProperties.rnc"
include "vml-officeDrawing.rnc"
include "vml-main.rnc"
include "vml-spreadsheetDrawing.rnc"
include "vml-wordprocessingDrawing.rnc"
include "shared-math.rnc"
```

```
start = w_fonts
```

B.1.1.5 Footer Part

This schema is available in the file WordprocessingML_Footer.rnc.

```
include "wml.rnc"
include "shared-relationshipReference.rnc"
include "dml-wordprocessingDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
include "vml-presentationDrawing.rnc"
include "xml.rnc"
include "shared-customXmlSchemaProperties.rnc"
include "vml-officeDrawing.rnc"
include "vml-main.rnc"
include "vml-spreadsheetDrawing.rnc"
include "vml-wordprocessingDrawing.rnc"
include "shared-math.rnc"
start = w_ftr
```

B.1.1.6 Footnotes Part

This schema is available in the file WordprocessingML_Footnotes.rnc.

```
include "wml.rnc"
include "shared-relationshipReference.rnc"
include "dml-wordprocessingDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
include "vml-presentationDrawing.rnc"
include "xml.rnc"
include "shared-customXmlSchemaProperties.rnc"
include "vml-officeDrawing.rnc"
include "vml-main.rnc"
include "vml-spreadsheetDrawing.rnc"
include "vml-wordprocessingDrawing.rnc"
include "shared-math.rnc"
start = w_footnotes
```


B.1.1.7 Glossary Document Part

This schema is available in the file WordprocessingML_Glossary_Document.rnc.

```
include "wml.rnc"
include "shared-relationshipReference.rnc"
include "dml-wordprocessingDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
include "vml-presentationDrawing.rnc"
include "xml.rnc"
include "shared-customXmlSchemaProperties.rnc"
include "vml-officeDrawing.rnc"
include "vml-main.rnc"
include "vml-spreadsheetDrawing.rnc"
include "vml-wordprocessingDrawing.rnc"
include "shared-math.rnc"
start = w_glossaryDocument
```

B.1.1.8 Header Part

This schema is available in the file WordprocessingML_Header.rnc.

```
include "wml.rnc"
include "shared-relationshipReference.rnc"
include "dml-wordprocessingDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
include "vml-presentationDrawing.rnc"
include "xml.rnc"
include "shared-customXmlSchemaProperties.rnc"
include "vml-officeDrawing.rnc"
include "vml-main.rnc"
include "vml-spreadsheetDrawing.rnc"
include "vml-wordprocessingDrawing.rnc"
include "shared-math.rnc"
start = w_hdr
```

B.1.1.9 Mail Merge Recipient Data Part

This schema is available in the file WordprocessingML_Mail_Merge_Recipient_Data.rnc.

```
include "wml.rnc"
include "shared-relationshipReference.rnc"
include "dml-wordprocessingDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
include "vml-presentationDrawing.rnc"
include "xml.rnc"
include "shared-customXmlSchemaProperties.rnc"
include "vml-officeDrawing.rnc"
include "vml-main.rnc"
include "vml-spreadsheetDrawing.rnc"
include "vml-wordprocessingDrawing.rnc"
include "shared-math.rnc"
start = w_recipients
```

B.1.1.10 Main Document Part

This schema is available in the file WordprocessingML_Main_Document.rnc.

```
include "wml.rnc"
include "shared-relationshipReference.rnc"
include "dml-wordprocessingDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
include "vml-presentationDrawing.rnc"
include "xml.rnc"
include "shared-customXmlSchemaProperties.rnc"
include "vml-officeDrawing.rnc"
include "vml-main.rnc"
include "vml-spreadsheetDrawing.rnc"
include "vml-wordprocessingDrawing.rnc"
include "shared-math.rnc"
start = w_document
```

B.1.1.11 Numbering Definitions Part

This schema is available in the file WordprocessingML_Numbering_Definitions.rnc.

```
include "wml.rnc"
include "shared-relationshipReference.rnc"
include "dml-wordprocessingDrawing.rnc"
include "dml-main.rnc"
```



```
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
include "vml-presentationDrawing.rnc"
include "xml.rnc"
include "shared-customXmlSchemaProperties.rnc"
include "vml-officeDrawing.rnc"
include "vml-main.rnc"
include "vml-spreadsheetDrawing.rnc"
include "vml-wordprocessingDrawing.rnc"
include "shared-math.rnc"
start = w_numbering
```

B.1.1.12 Style Definitions Part

This schema is available in the file WordprocessingML_Style_Definitions.rnc.

```
include "wml.rnc"
include "shared-relationshipReference.rnc"
include "dml-wordprocessingDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
include "vml-presentationDrawing.rnc"
include "xml.rnc"
include "shared-customXmlSchemaProperties.rnc"
include "vml-officeDrawing.rnc"
include "vml-main.rnc"
include "vml-spreadsheetDrawing.rnc"
include "vml-wordprocessingDrawing.rnc"
include "shared-math.rnc"
start = w_styles
```

B.1.1.13 Web Settings Part

This schema is available in the file WordprocessingML_Web_Settings.rnc.

```
include "wml.rnc"
include "shared-relationshipReference.rnc"
include "dml-wordprocessingDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
```

```
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
include "vml-presentationDrawing.rnc"
include "xml.rnc"
include "shared-customXmlSchemaProperties.rnc"
include "vml-officeDrawing.rnc"
include "vml-main.rnc"
include "vml-spreadsheetDrawing.rnc"
include "vml-wordprocessingDrawing.rnc"
include "shared-math.rnc"
start = w_webSettings
```

B.2 SpreadsheetML

This schema is available in the file sml.rnc.

```
namespace o = "urn:schemas-microsoft-com:office:office"
namespace r =
  "http://schemas.openxmlformats.org/officeDocument/2006/relationships"
namespace s =
  "http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
default namespace sml =
  "http://schemas.openxmlformats.org/spreadsheetml/2006/main"
namespace v = "urn:schemas-microsoft-com:vml"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"
namespace xdr =
  "http://schemas.openxmlformats.org/drawingml/2006/spreadsheetDrawing"

sml_CT_AutoFilter =
  attribute ref { sml_ST_Ref }?,
  element filterColumn { sml_CT_FilterColumn }*,
  element sortState { sml_CT_SortState }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_FilterColumn =
  attribute colId { xsd:unsignedInt },

  ## default value: false
  attribute hiddenButton { xsd:boolean }?,

  ## default value: true
  attribute showButton { xsd:boolean }?,
  (element filters { sml_CT_Filters }?
   | element top10 { sml_CT_Top10 }?
   | element customFilters { sml_CT_CustomFilters }?
   | element dynamicFilter { sml_CT_DynamicFilter }?
   | element colorFilter { sml_CT_ColorFilter }?
   | element iconFilter { sml_CT_IconFilter }?
   | element extLst { sml_CT_ExtensionList }?)?
sml_CT_Filters =

  ## default value: false
```



```
  attribute blank { xsd:boolean }?,

  ## default value: none
  attribute calendarType { s_ST_CalendarType }?,
  element filter { sml_CT_Filter }*,
  element dateGroupItem { sml_CT_DateGroupItem }*
sml_CT_Filter = attribute val { s_ST_Xstring }?
sml_CT_CustomFilters =

  ## default value: false
  attribute and { xsd:boolean }?,
  element customFilter { sml_CT_CustomFilter }+
sml_CT_CustomFilter =

  ## default value: equal
  attribute operator { sml_ST_FilterOperator }?,
  attribute val { s_ST_Xstring }?
sml_CT_Top10 =

  ## default value: true
  attribute top { xsd:boolean }?,

  ## default value: false
  attribute percent { xsd:boolean }?,
  attribute val { xsd:double },
  attribute filterVal { xsd:double }?
sml_CT_ColorFilter =
  attribute dxfId { sml_ST_DxfId }?,

  ## default value: true
  attribute cellColor { xsd:boolean }?
sml_CT_IconFilter =
  attribute iconSet { sml_ST_IconSetType },
  attribute iconId { xsd:unsignedInt }?
sml_ST_FilterOperator =
  string "equal"
  | string "lessThan"
  | string "lessThanOrEqual"
  | string "notEqual"
  | string "greaterThanOrEqual"
  | string "greaterThan"
sml_CT_DynamicFilter =
  attribute type { sml_ST_DynamicFilterType },
  attribute val { xsd:double }?,
  attribute valIso { xsd:dateTime }?,
  attribute maxVal { xsd:double }?,
  attribute maxValIso { xsd:dateTime }?
sml_ST_DynamicFilterType =
  string "null"
  | string "aboveAverage"
  | string "belowAverage"
  | string "tomorrow"
  | string "today"
```

```
  | string "yesterday"
  | string "nextWeek"
  | string "thisWeek"
  | string "lastWeek"
  | string "nextMonth"
  | string "thisMonth"
  | string "lastMonth"
  | string "nextQuarter"
  | string "thisQuarter"
  | string "lastQuarter"
  | string "nextYear"
  | string "thisYear"
  | string "lastYear"
  | string "yearToDate"
  | string "Q1"
  | string "Q2"
  | string "Q3"
  | string "Q4"
  | string "M1"
  | string "M2"
  | string "M3"
  | string "M4"
  | string "M5"
  | string "M6"
  | string "M7"
  | string "M8"
  | string "M9"
  | string "M10"
  | string "M11"
  | string "M12"
sml_ST_IconSetType =
  string "3Arrows"
  | string "3ArrowsGray"
  | string "3Flags"
  | string "3TrafficLights1"
  | string "3TrafficLights2"
  | string "3Signs"
  | string "3Symbols"
  | string "3Symbols2"
  | string "4Arrows"
  | string "4ArrowsGray"
  | string "4RedToBlack"
  | string "4Rating"
  | string "4TrafficLights"
  | string "5Arrows"
  | string "5ArrowsGray"
  | string "5Rating"
  | string "5Quarters"
sml_CT_SortState =

  ## default value: false
  attribute columnSort { xsd:boolean }?,

```



```
  ## default value: false
  attribute caseSensitive { xsd:boolean }?,

  ## default value: none
  attribute sortMethod { sml_ST_SortMethod }?,
  attribute ref { sml_ST_Ref },
  element sortCondition { sml_CT_SortCondition }*,
  element extLst { sml_CT_ExtensionList }?
sml_CT_SortCondition =

  ## default value: false
  attribute descending { xsd:boolean }?,

  ## default value: value
  attribute sortBy { sml_ST_SortBy }?,
  attribute ref { sml_ST_Ref },
  attribute customList { s_ST_Xstring }?,
  attribute dxfId { sml_ST_DxfId }?,

  ## default value: 3Arrows
  attribute iconSet { sml_ST_IconSetType }?,
  attribute iconId { xsd:unsignedInt }?
sml_ST_SortBy =
  string "value"
  | string "cellColor"
  | string "fontColor"
  | string "icon"
sml_ST_SortMethod = string "stroke" | string "pinYin" | string "none"
sml_CT_DateGroupItem =
  attribute year { xsd:unsignedShort },
  attribute month { xsd:unsignedShort }?,
  attribute day { xsd:unsignedShort }?,
  attribute hour { xsd:unsignedShort }?,
  attribute minute { xsd:unsignedShort }?,
  attribute second { xsd:unsignedShort }?,
  attribute dateTimeGrouping { sml_ST_DateTimeGrouping }
sml_ST_DateTimeGrouping =
  string "year"
  | string "month"
  | string "day"
  | string "hour"
  | string "minute"
  | string "second"
sml_ST_CellRef = xsd:string
sml_ST_Ref = xsd:string
sml_ST_RefA = xsd:string
sml_ST_Sqref = list { sml_ST_Ref* }
sml_ST_Formula = s_ST_Xstring
sml_ST_UnsignedIntHex = xsd:hexBinary { length = "4" }
sml_ST_UnsignedShortHex = xsd:hexBinary { length = "2" }
sml_CT_XStringElement = attribute v { s_ST_Xstring }
sml_CT_Extension =
  attribute uri { xsd:token }?,
```

```
  sml_CT_Extension_any
sml_CT_Extension_any =
  element * - (o:* | v:* | w10:* | x:*) {
    anyAttribute*,
    mixed { anyElement* }
  }
sml_CT_ObjectAnchor =

  ## default value: false
  attribute moveWithCells { xsd:boolean }?,

  ## default value: false
  attribute sizeWithCells { xsd:boolean }?,
  xdr_from,
  xdr_to
sml_EG_ExtensionList = element ext { sml_CT_Extension }*
sml_CT_ExtensionList = sml_EG_ExtensionList?
sml_calcChain = element calcChain { sml_CT_CalcChain }
sml_CT_CalcChain =
  element c { sml_CT_CalcCell }+,
  element extLst { sml_CT_ExtensionList }?
sml_CT_CalcCell =
  attribute ( r | ref ) { sml_ST_CellRef },

  ## default value: 0
  attribute i { xsd:int }?,

  ## default value: false
  attribute s { xsd:boolean }?,

  ## default value: false
  attribute l { xsd:boolean }?,

  ## default value: false
  attribute t { xsd:boolean }?,

  ## default value: false
  attribute a { xsd:boolean }?
sml_comments = element comments { sml_CT_Comments }
sml_CT_Comments =
  element authors { sml_CT_Authors },
  element commentList { sml_CT_CommentList },
  element extLst { sml_CT_ExtensionList }?
sml_CT_Authors = element author { s_ST_Xstring }*
sml_CT_CommentList = element comment { sml_CT_Comment }*
sml_CT_Comment =
  attribute ref { sml_ST_Ref },
  attribute authorId { xsd:unsignedInt },
  attribute guid { s_ST_Guid }?,
  attribute shapeId { xsd:unsignedInt }?,
  element text { sml_CT_Rst },
  element commentPr { sml_CT_CommentPr }?
sml_CT_CommentPr =
```



```
  ## default value: true
  attribute locked { xsd:boolean }?,

  ## default value: true
  attribute defaultSize { xsd:boolean }?,

  ## default value: true
  attribute print { xsd:boolean }?,

  ## default value: false
  attribute disabled { xsd:boolean }?,

  ## default value: true
  attribute autoFill { xsd:boolean }?,

  ## default value: true
  attribute autoLine { xsd:boolean }?,
  attribute altText { s_ST_Xstring }?,

  ## default value: left
  attribute textHAlign { sml_ST_TextHAlign }?,

  ## default value: top
  attribute textVAlign { sml_ST_TextVAlign }?,

  ## default value: true
  attribute lockText { xsd:boolean }?,

  ## default value: false
  attribute justLastX { xsd:boolean }?,

  ## default value: false
  attribute autoScale { xsd:boolean }?,
  element anchor { sml_CT_ObjectAnchor }
sml_ST_TextHAlign =
  string "left"
  | string "center"
  | string "right"
  | string "justify"
  | string "distributed"
sml_ST_TextVAlign =
  string "top"
  | string "center"
  | string "bottom"
  | string "justify"
  | string "distributed"
sml_MapInfo = element MapInfo { sml_CT_MapInfo }
sml_CT_MapInfo =
  attribute SelectionNamespaces { xsd:string },
  element Schema { sml_CT_Schema }+,
  element Map { sml_CT_Map }+
sml_CT_Schema =
```

```
  mixed {
    attribute ID { xsd:string },
    attribute SchemaRef { xsd:string }?,
    attribute Namespace { xsd:string }?,
    attribute SchemaLanguage { xsd:token }?,
    sml_CT_Schema_any
  }
sml_CT_Schema_any =
  element * - (o:* | v:* | w10:* | x:*) {
    anyAttribute*,
    mixed { anyElement* }
  }
sml_CT_Map =
  attribute ID { xsd:unsignedInt },
  attribute Name { xsd:string },
  attribute RootElement { xsd:string },
  attribute SchemaID { xsd:string },
  attribute ShowImportExportValidationErrors { xsd:boolean },
  attribute AutoFit { xsd:boolean },
  attribute Append { xsd:boolean },
  attribute PreserveSortAFLayout { xsd:boolean },
  attribute PreserveFormat { xsd:boolean },
  element DataBinding { sml_CT_DataBinding }?
sml_CT_DataBinding =
  attribute DataBindingName { xsd:string }?,
  attribute FileBinding { xsd:boolean }?,
  attribute ConnectionID { xsd:unsignedInt }?,
  attribute FileBindingName { xsd:string }?,
  attribute DataBindingLoadMode { xsd:unsignedInt },
  sml_CT_DataBinding_any
sml_CT_DataBinding_any =
  element * - (o:* | v:* | w10:* | x:*) {
    anyAttribute*,
    mixed { anyElement* }
  }
sml_connections = element connections { sml_CT_Connections }
sml_CT_Connections = element connection { sml_CT_Connection }+
sml_CT_Connection =
  attribute id { xsd:unsignedInt },
  attribute sourceFile { s_ST_Xstring }?,
  attribute odcFile { s_ST_Xstring }?,

  ## default value: false
  attribute keepAlive { xsd:boolean }?,

  ## default value: 0
  attribute interval { xsd:unsignedInt }?,
  attribute name { s_ST_Xstring }?,
  attribute description { s_ST_Xstring }?,
  attribute type { xsd:unsignedInt }?,

  ## default value: 1
  attribute reconnectionMethod { xsd:unsignedInt }?,
```



```
  attribute refreshedVersion { xsd:unsignedByte },

  ## default value: 0
  attribute minRefreshableVersion { xsd:unsignedByte }?,

  ## default value: false
  attribute savePassword { xsd:boolean }?,

  ## default value: false
  attribute new { xsd:boolean }?,

  ## default value: false
  attribute deleted { xsd:boolean }?,

  ## default value: false
  attribute onlyUseConnectionFile { xsd:boolean }?,

  ## default value: false
  attribute background { xsd:boolean }?,

  ## default value: false
  attribute refreshOnLoad { xsd:boolean }?,

  ## default value: false
  attribute saveData { xsd:boolean }?,

  ## default value: integrated
  attribute credentials { sml_ST_CredMethod }?,
  attribute singleSignOnId { s_ST_Xstring }?,
  element dbPr { sml_CT_DbPr }?,
  element olapPr { sml_CT_OlapPr }?,
  element webPr { sml_CT_WebPr }?,
  element textPr { sml_CT_TextPr }?,
  element parameters { sml_CT_Parameters }?,
  element extLst { sml_CT_ExtensionList }?
sml_ST_CredMethod =
  string "integrated"
  | string "none"
  | string "stored"
  | string "prompt"
sml_CT_DbPr =
  attribute connection { s_ST_Xstring },
  attribute command { s_ST_Xstring }?,
  attribute serverCommand { s_ST_Xstring }?,

  ## default value: 2
  attribute commandType { xsd:unsignedInt }?
sml_CT_OlapPr =

  ## default value: false
  attribute local { xsd:boolean }?,
  attribute localConnection { s_ST_Xstring }?,

```

```
  ## default value: true
  attribute localRefresh { xsd:boolean }?,

  ## default value: false
  attribute sendLocale { xsd:boolean }?,
  attribute rowDrillCount { xsd:unsignedInt }?,

  ## default value: true
  attribute serverFill { xsd:boolean }?,

  ## default value: true
  attribute serverNumberFormat { xsd:boolean }?,

  ## default value: true
  attribute serverFont { xsd:boolean }?,

  ## default value: true
  attribute serverFontColor { xsd:boolean }?
sml_CT_WebPr =

  ## default value: false
  attribute xml { xsd:boolean }?,

  ## default value: false
  attribute sourceData { xsd:boolean }?,

  ## default value: false
  attribute parsePre { xsd:boolean }?,

  ## default value: false
  attribute consecutive { xsd:boolean }?,

  ## default value: false
  attribute firstRow { xsd:boolean }?,

  ## default value: false
  attribute xl97 { xsd:boolean }?,

  ## default value: false
  attribute textDates { xsd:boolean }?,

  ## default value: false
  attribute xl2000 { xsd:boolean }?,
  attribute url { s_ST_Xstring }?,
  attribute post { s_ST_Xstring }?,

  ## default value: false
  attribute htmlTables { xsd:boolean }?,

  ## default value: none
  attribute htmlFormat { sml_ST_HtmlFmt }?,
  attribute editPage { s_ST_Xstring }?,
  element tables { sml_CT_Tables }?
```



```
sml_ST_HtmlFmt = string "none" | string "rtf" | string "all"
sml_CT_Parameters =
  attribute count { xsd:unsignedInt }?,
  element parameter { sml_CT_Parameter }+
sml_CT_Parameter =
  attribute name { s_ST_Xstring }?,

  ## default value: 0
  attribute sqlType { xsd:int }?,

  ## default value: prompt
  attribute parameterType { sml_ST_ParameterType }?,

  ## default value: false
  attribute refreshOnChange { xsd:boolean }?,
  attribute prompt { s_ST_Xstring }?,
  attribute boolean { xsd:boolean }?,
  attribute double { xsd:double }?,
  attribute integer { xsd:int }?,
  attribute string { s_ST_Xstring }?,
  attribute cell { s_ST_Xstring }?
sml_ST_ParameterType = string "prompt" | string "value" | string "cell"
sml_CT_Tables =
  attribute count { xsd:unsignedInt }?,
  (element m { sml_CT_TableMissing }
   | element s { sml_CT_XStringElement }
   | element x { sml_CT_Index })+
sml_CT_TableMissing = empty
sml_CT_TextPr =

  ## default value: true
  attribute prompt { xsd:boolean }?,

  ## default value: win
  attribute fileType { sml_ST_FileType }?,

  ## default value: 1252
  attribute codePage { xsd:unsignedInt }?,
  attribute characterSet { xsd:string }?,

  ## default value: 1
  attribute firstRow { xsd:unsignedInt }?,
  attribute sourceFile { s_ST_Xstring }?,

  ## default value: true
  attribute delimited { xsd:boolean }?,

  ## default value: .
  attribute decimal { s_ST_Xstring }?,

  ## default value: ,
  attribute thousands { s_ST_Xstring }?,

```

```
  ## default value: true
  attribute tab { xsd:boolean }?,

  ## default value: false
  attribute space { xsd:boolean }?,

  ## default value: false
  attribute comma { xsd:boolean }?,

  ## default value: false
  attribute semicolon { xsd:boolean }?,

  ## default value: false
  attribute consecutive { xsd:boolean }?,

  ## default value: doubleQuote
  attribute qualifier { sml_ST_Qualifier }?,
  attribute delimiter { s_ST_Xstring }?,
  element textFields { sml_CT_TextFields }?
sml_ST_FileType =
  string "mac"
  | string "win"
  | string "dos"
  | string "lin"
  | string "other"
sml_ST_Qualifier =
  string "doubleQuote" | string "singleQuote" | string "none"
sml_CT_TextFields =

  ## default value: 1
  attribute count { xsd:unsignedInt }?,
  element textField { sml_CT_TextField }+
sml_CT_TextField =

  ## default value: general
  attribute type { sml_ST_ExternalConnectionType }?,

  ## default value: 0
  attribute position { xsd:unsignedInt }?
sml_ST_ExternalConnectionType =
  string "general"
  | string "text"
  | string "MDY"
  | string "DMY"
  | string "YMD"
  | string "MYD"
  | string "DYM"
  | string "YDM"
  | string "skip"
  | string "EMD"
sml_pivotCacheDefinition =
  element pivotCacheDefinition { sml_CT_PivotCacheDefinition }
sml_pivotCacheRecords =
```



```
  element pivotCacheRecords { sml_CT_PivotCacheRecords }
sml_pivotTableDefinition =
  element pivotTableDefinition { sml_CT_pivotTableDefinition }
sml_CT_PivotCacheDefinition =
  r_id?,

  ## default value: false
  attribute invalid { xsd:boolean }?,

  ## default value: true
  attribute saveData { xsd:boolean }?,

  ## default value: false
  attribute refreshOnLoad { xsd:boolean }?,

  ## default value: false
  attribute optimizeMemory { xsd:boolean }?,

  ## default value: true
  attribute enableRefresh { xsd:boolean }?,
  attribute refreshedBy { s_ST_Xstring }?,
  attribute refreshedDate { xsd:double }?,
  attribute refreshedDateIso { xsd:dateTime }?,

  ## default value: false
  attribute backgroundQuery { xsd:boolean }?,
  attribute missingItemsLimit { xsd:unsignedInt }?,

  ## default value: 0
  attribute createdVersion { xsd:unsignedByte }?,

  ## default value: 0
  attribute refreshedVersion { xsd:unsignedByte }?,

  ## default value: 0
  attribute minRefreshableVersion { xsd:unsignedByte }?,
  attribute recordCount { xsd:unsignedInt }?,

  ## default value: false
  attribute upgradeOnRefresh { xsd:boolean }?,

  ## default value: false
  attribute tupleCache { xsd:boolean }?,

  ## default value: false
  attribute supportSubquery { xsd:boolean }?,

  ## default value: false
  attribute supportAdvancedDrill { xsd:boolean }?,
  element cacheSource { sml_CT_CacheSource },
  element cacheFields { sml_CT_CacheFields },
  element cacheHierarchies { sml_CT_CacheHierarchies }?,
  element kpis { sml_CT_PCDKPIs }?,
```

```
  element tupleCache { sml_CT_TupleCache }?,
  element calculatedItems { sml_CT_CalculatedItems }?,
  element calculatedMembers { sml_CT_CalculatedMembers }?,
  element dimensions { sml_CT_Dimensions }?,
  element measureGroups { sml_CT_MeasureGroups }?,
  element maps { sml_CT_MeasureDimensionMaps }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_CacheFields =
  attribute count { xsd:unsignedInt }?,
  element cacheField { sml_CT_CacheField }*
sml_CT_CacheField =
  attribute name { s_ST_Xstring },
  attribute caption { s_ST_Xstring }?,
  attribute propertyName { s_ST_Xstring }?,

  ## default value: false
  attribute serverField { xsd:boolean }?,

  ## default value: true
  attribute uniqueList { xsd:boolean }?,
  attribute numFmtId { sml_ST_NumFmtId }?,
  attribute formula { s_ST_Xstring }?,

  ## default value: 0
  attribute sqlType { xsd:int }?,

  ## default value: 0
  attribute hierarchy { xsd:int }?,

  ## default value: 0
  attribute level { xsd:unsignedInt }?,

  ## default value: true
  attribute databaseField { xsd:boolean }?,
  attribute mappingCount { xsd:unsignedInt }?,

  ## default value: false
  attribute memberPropertyField { xsd:boolean }?,
  element sharedItems { sml_CT_SharedItems }?,
  element fieldGroup { sml_CT_FieldGroup }?,
  element mpMap { sml_CT_X }*,
  element extLst { sml_CT_ExtensionList }?
sml_CT_CacheSource =
  attribute type { sml_ST_SourceType },

  ## default value: 0
  attribute connectionId { xsd:unsignedInt }?,
  (element worksheetSource { sml_CT_WorksheetSource }
   | element consolidation { sml_CT_Consolidation }
   | element extLst { sml_CT_ExtensionList }?)?
sml_ST_SourceType =
  string "worksheet"
  | string "external"
```



```
  | string "consolidation"
  | string "scenario"
sml_CT_WorksheetSource =
  attribute ref { sml_ST_Ref }?,
  attribute name { s_ST_Xstring }?,
  attribute sheet { s_ST_Xstring }?,
  r_id?
sml_CT_Consolidation =

  ## default value: true
  attribute autoPage { xsd:boolean }?,
  element pages { sml_CT_Pages }?,
  element rangeSets { sml_CT_RangeSets }
sml_CT_Pages =
  attribute count { xsd:unsignedInt }?,
  element page { sml_CT_PCDSCPage }+
sml_CT_PCDSCPage =
  attribute count { xsd:unsignedInt }?,
  element pageItem { sml_CT_PageItem }*
sml_CT_PageItem = attribute name { s_ST_Xstring }
sml_CT_RangeSets =
  attribute count { xsd:unsignedInt }?,
  element rangeSet { sml_CT_RangeSet }+
sml_CT_RangeSet =
  attribute i1 { xsd:unsignedInt }?,
  attribute i2 { xsd:unsignedInt }?,
  attribute i3 { xsd:unsignedInt }?,
  attribute i4 { xsd:unsignedInt }?,
  attribute ref { sml_ST_Ref }?,
  attribute name { s_ST_Xstring }?,
  attribute sheet { s_ST_Xstring }?,
  r_id?
sml_CT_SharedItems =

  ## default value: true
  attribute containsSemiMixedTypes { xsd:boolean }?,

  ## default value: true
  attribute containsNonDate { xsd:boolean }?,

  ## default value: false
  attribute containsDate { xsd:boolean }?,

  ## default value: true
  attribute containsString { xsd:boolean }?,

  ## default value: false
  attribute containsBlank { xsd:boolean }?,

  ## default value: false
  attribute containsMixedTypes { xsd:boolean }?,

  ## default value: false
```

```
  attribute containsNumber { xsd:boolean }?,

  ## default value: false
  attribute containsInteger { xsd:boolean }?,
  attribute minValue { xsd:double }?,
  attribute maxValue { xsd:double }?,
  attribute minDate { xsd:dateTime }?,
  attribute maxDate { xsd:dateTime }?,
  attribute count { xsd:unsignedInt }?,

  ## default value: false
  attribute longText { xsd:boolean }?,
  (element m { sml_CT_Missing }
   | element n { sml_CT_Number }
   | element b { sml_CT_Boolean }
   | element e { sml_CT_Error }
   | element s { sml_CT_String }
   | element d { sml_CT_DateTime })*
sml_CT_Missing =
  attribute u { xsd:boolean }?,
  attribute f { xsd:boolean }?,
  attribute c { s_ST_Xstring }?,
  attribute cp { xsd:unsignedInt }?,
  attribute in { xsd:unsignedInt }?,
  attribute bc { sml_ST_UnsignedIntHex }?,
  attribute fc { sml_ST_UnsignedIntHex }?,

  ## default value: false
  attribute i { xsd:boolean }?,

  ## default value: false
  attribute un { xsd:boolean }?,

  ## default value: false
  attribute st { xsd:boolean }?,

  ## default value: false
  attribute b { xsd:boolean }?,
  element tpls { sml_CT_Tuples }*,
  element x { sml_CT_X }*
sml_CT_Number =
  attribute v { xsd:double },
  attribute u { xsd:boolean }?,
  attribute f { xsd:boolean }?,
  attribute c { s_ST_Xstring }?,
  attribute cp { xsd:unsignedInt }?,
  attribute in { xsd:unsignedInt }?,
  attribute bc { sml_ST_UnsignedIntHex }?,
  attribute fc { sml_ST_UnsignedIntHex }?,

  ## default value: false
  attribute i { xsd:boolean }?,

```



```
  ## default value: false
  attribute un { xsd:boolean }?,

  ## default value: false
  attribute st { xsd:boolean }?,

  ## default value: false
  attribute b { xsd:boolean }?,
  element tpls { sml_CT_Tuples }*,
  element x { sml_CT_X }*
sml_CT_Boolean =
  attribute v { xsd:boolean },
  attribute u { xsd:boolean }?,
  attribute f { xsd:boolean }?,
  attribute c { s_ST_Xstring }?,
  attribute cp { xsd:unsignedInt }?,
  element x { sml_CT_X }*
sml_CT_Error =
  attribute v { s_ST_Xstring },
  attribute u { xsd:boolean }?,
  attribute f { xsd:boolean }?,
  attribute c { s_ST_Xstring }?,
  attribute cp { xsd:unsignedInt }?,
  attribute in { xsd:unsignedInt }?,
  attribute bc { sml_ST_UnsignedIntHex }?,
  attribute fc { sml_ST_UnsignedIntHex }?,

  ## default value: false
  attribute i { xsd:boolean }?,

  ## default value: false
  attribute un { xsd:boolean }?,

  ## default value: false
  attribute st { xsd:boolean }?,

  ## default value: false
  attribute b { xsd:boolean }?,
  element tpls { sml_CT_Tuples }?,
  element x { sml_CT_X }*
sml_CT_String =
  attribute v { s_ST_Xstring },
  attribute u { xsd:boolean }?,
  attribute f { xsd:boolean }?,
  attribute c { s_ST_Xstring }?,
  attribute cp { xsd:unsignedInt }?,
  attribute in { xsd:unsignedInt }?,
  attribute bc { sml_ST_UnsignedIntHex }?,
  attribute fc { sml_ST_UnsignedIntHex }?,

  ## default value: false
  attribute i { xsd:boolean }?,

```

```
  ## default value: false
  attribute un { xsd:boolean }?,

  ## default value: false
  attribute st { xsd:boolean }?,

  ## default value: false
  attribute b { xsd:boolean }?,
  element tpls { sml_CT_Tuples }*,
  element x { sml_CT_X }*
sml_CT_DateTime =
  attribute v { xsd:dateTime },
  attribute u { xsd:boolean }?,
  attribute f { xsd:boolean }?,
  attribute c { s_ST_Xstring }?,
  attribute cp { xsd:unsignedInt }?,
  element x { sml_CT_X }*
sml_CT_FieldGroup =
  attribute par { xsd:unsignedInt }?,
  attribute base { xsd:unsignedInt }?,
  element rangePr { sml_CT_RangePr }?,
  element discretePr { sml_CT_DiscretePr }?,
  element groupItems { sml_CT_GroupItems }?
sml_CT_RangePr =

  ## default value: true
  attribute autoStart { xsd:boolean }?,

  ## default value: true
  attribute autoEnd { xsd:boolean }?,

  ## default value: range
  attribute groupBy { sml_ST_GroupBy }?,
  attribute startNum { xsd:double }?,
  attribute endNum { xsd:double }?,
  attribute startDate { xsd:dateTime }?,
  attribute endDate { xsd:dateTime }?,

  ## default value: 1
  attribute groupInterval { xsd:double }?
sml_ST_GroupBy =
  string "range"
  | string "seconds"
  | string "minutes"
  | string "hours"
  | string "days"
  | string "months"
  | string "quarters"
  | string "years"
sml_CT_DiscretePr =
  attribute count { xsd:unsignedInt }?,
  element x { sml_CT_Index }+
sml_CT_GroupItems =
```



```
  attribute count { xsd:unsignedInt }?,
  (element m { sml_CT_Missing }
   | element n { sml_CT_Number }
   | element b { sml_CT_Boolean }
   | element e { sml_CT_Error }
   | element s { sml_CT_String }
   | element d { sml_CT_DateTime })+
sml_CT_PivotCacheRecords =
  attribute count { xsd:unsignedInt }?,
  element r { sml_CT_Record }*,
  element extLst { sml_CT_ExtensionList }?
sml_CT_Record =
  (element m { sml_CT_Missing }
   | element n { sml_CT_Number }
   | element b { sml_CT_Boolean }
   | element e { sml_CT_Error }
   | element s { sml_CT_String }
   | element d { sml_CT_DateTime }
   | element x { sml_CT_Index })+
sml_CT_PCDKPIs =
  attribute count { xsd:unsignedInt }?,
  element kpi { sml_CT_PCDKPI }*
sml_CT_PCDKPI =
  attribute uniqueName { s_ST_Xstring },
  attribute caption { s_ST_Xstring }?,
  attribute displayFolder { s_ST_Xstring }?,
  attribute measureGroup { s_ST_Xstring }?,
  attribute parent { s_ST_Xstring }?,
  attribute value { s_ST_Xstring },
  attribute goal { s_ST_Xstring }?,
  attribute status { s_ST_Xstring }?,
  attribute trend { s_ST_Xstring }?,
  attribute weight { s_ST_Xstring }?,
  attribute time { s_ST_Xstring }?
sml_CT_CacheHierarchies =
  attribute count { xsd:unsignedInt }?,
  element cacheHierarchy { sml_CT_CacheHierarchy }*
sml_CT_CacheHierarchy =
  attribute uniqueName { s_ST_Xstring },
  attribute caption { s_ST_Xstring }?,

  ## default value: false
  attribute measure { xsd:boolean }?,

  ## default value: false
  attribute set { xsd:boolean }?,
  attribute parentSet { xsd:unsignedInt }?,

  ## default value: 0
  attribute iconSet { xsd:int }?,

  ## default value: false
  attribute attribute { xsd:boolean }?,
```

```
  ## default value: false
  attribute time { xsd:boolean }?,

  ## default value: false
  attribute keyAttribute { xsd:boolean }?,
  attribute defaultMemberUniqueName { s_ST_Xstring }?,
  attribute allUniqueName { s_ST_Xstring }?,
  attribute allCaption { s_ST_Xstring }?,
  attribute dimensionUniqueName { s_ST_Xstring }?,
  attribute displayFolder { s_ST_Xstring }?,
  attribute measureGroup { s_ST_Xstring }?,

  ## default value: false
  attribute measures { xsd:boolean }?,
  attribute count { xsd:unsignedInt },

  ## default value: false
  attribute oneField { xsd:boolean }?,
  attribute memberValueDatatype { xsd:unsignedShort }?,
  attribute unbalanced { xsd:boolean }?,
  attribute unbalancedGroup { xsd:boolean }?,

  ## default value: false
  attribute hidden { xsd:boolean }?,
  element fieldsUsage { sml_CT_FieldsUsage }?,
  element groupLevels { sml_CT_GroupLevels }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_FieldsUsage =
  attribute count { xsd:unsignedInt }?,
  element fieldUsage { sml_CT_FieldUsage }*
sml_CT_FieldUsage = attribute x { xsd:int }
sml_CT_GroupLevels =
  attribute count { xsd:unsignedInt }?,
  element groupLevel { sml_CT_GroupLevel }+
sml_CT_GroupLevel =
  attribute uniqueName { s_ST_Xstring },
  attribute caption { s_ST_Xstring },

  ## default value: false
  attribute user { xsd:boolean }?,

  ## default value: false
  attribute customRollUp { xsd:boolean }?,
  element groups { sml_CT_Groups }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_Groups =
  attribute count { xsd:unsignedInt }?,
  element group { sml_CT_LevelGroup }+
sml_CT_LevelGroup =
  attribute name { s_ST_Xstring },
  attribute uniqueName { s_ST_Xstring },
  attribute caption { s_ST_Xstring },
```



```
  attribute uniqueParent { s_ST_Xstring }?,
  attribute id { xsd:int }?,
  element groupMembers { sml_CT_GroupMembers }
sml_CT_GroupMembers =
  attribute count { xsd:unsignedInt }?,
  element groupMember { sml_CT_GroupMember }+
sml_CT_GroupMember =
  attribute uniqueName { s_ST_Xstring },

  ## default value: false
  attribute group { xsd:boolean }?
sml_CT_TupleCache =
  element entries { sml_CT_PCDSDTCEntries }?,
  element sets { sml_CT_Sets }?,
  element queryCache { sml_CT_QueryCache }?,
  element serverFormats { sml_CT_ServerFormats }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_ServerFormat =
  attribute culture { s_ST_Xstring }?,
  attribute format { s_ST_Xstring }?
sml_CT_ServerFormats =
  attribute count { xsd:unsignedInt }?,
  element serverFormat { sml_CT_ServerFormat }*
sml_CT_PCDSDTCEntries =
  attribute count { xsd:unsignedInt }?,
  (element m { sml_CT_Missing }
   | element n { sml_CT_Number }
   | element e { sml_CT_Error }
   | element s { sml_CT_String })+
sml_CT_Tuples =
  attribute c { xsd:unsignedInt }?,
  element tpl { sml_CT_Tuple }+
sml_CT_Tuple =
  attribute fld { xsd:unsignedInt }?,
  attribute hier { xsd:unsignedInt }?,
  attribute item { xsd:unsignedInt }
sml_CT_Sets =
  attribute count { xsd:unsignedInt }?,
  element set { sml_CT_Set }+
sml_CT_Set =
  attribute count { xsd:unsignedInt }?,
  attribute maxRank { xsd:int },
  attribute setDefinition { s_ST_Xstring },

  ## default value: none
  attribute sortType { sml_ST_SortType }?,

  ## default value: false
  attribute queryFailed { xsd:boolean }?,
  element tpls { sml_CT_Tuples }*,
  element sortByTuple { sml_CT_Tuples }?
sml_ST_SortType =
  string "none"
```

```
  | string "ascending"
  | string "descending"
  | string "ascendingAlpha"
  | string "descendingAlpha"
  | string "ascendingNatural"
  | string "descendingNatural"
sml_CT_QueryCache =
  attribute count { xsd:unsignedInt }?,
  element query { sml_CT_Query }+
sml_CT_Query =
  attribute mdx { s_ST_Xstring },
  element tpls { sml_CT_Tuples }?
sml_CT_CalculatedItems =
  attribute count { xsd:unsignedInt }?,
  element calculatedItem { sml_CT_CalculatedItem }+
sml_CT_CalculatedItem =
  attribute field { xsd:unsignedInt }?,
  attribute formula { s_ST_Xstring }?,
  element pivotArea { sml_CT_PivotArea },
  element extLst { sml_CT_ExtensionList }?
sml_CT_CalculatedMembers =
  attribute count { xsd:unsignedInt }?,
  element calculatedMember { sml_CT_CalculatedMember }+
sml_CT_CalculatedMember =
  attribute name { s_ST_Xstring },
  attribute mdx { s_ST_Xstring },
  attribute memberName { s_ST_Xstring }?,
  attribute hierarchy { s_ST_Xstring }?,
  attribute parent { s_ST_Xstring }?,

  ## default value: 0
  attribute solveOrder { xsd:int }?,

  ## default value: false
  attribute set { xsd:boolean }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_pivotTableDefinition =
  attribute name { s_ST_Xstring },
  attribute cacheId { xsd:unsignedInt },

  ## default value: false
  attribute dataOnRows { xsd:boolean }?,
  attribute dataPosition { xsd:unsignedInt }?,
  sml_AG_AutoFormat,
  attribute dataCaption { s_ST_Xstring },
  attribute grandTotalCaption { s_ST_Xstring }?,
  attribute errorCaption { s_ST_Xstring }?,

  ## default value: false
  attribute showError { xsd:boolean }?,
  attribute missingCaption { s_ST_Xstring }?,

  ## default value: true
```



```
  attribute showMissing { xsd:boolean }?,
  attribute pageStyle { s_ST_Xstring }?,
  attribute pivotTableStyle { s_ST_Xstring }?,
  attribute vacatedStyle { s_ST_Xstring }?,
  attribute tag { s_ST_Xstring }?,

  ## default value: 0
  attribute updatedVersion { xsd:unsignedByte }?,

  ## default value: 0
  attribute minRefreshableVersion { xsd:unsignedByte }?,

  ## default value: false
  attribute asteriskTotals { xsd:boolean }?,

  ## default value: true
  attribute showItems { xsd:boolean }?,

  ## default value: false
  attribute editData { xsd:boolean }?,

  ## default value: false
  attribute disableFieldList { xsd:boolean }?,

  ## default value: true
  attribute showCalcMbrs { xsd:boolean }?,

  ## default value: true
  attribute visualTotals { xsd:boolean }?,

  ## default value: true
  attribute showMultipleLabel { xsd:boolean }?,

  ## default value: true
  attribute showDataDropDown { xsd:boolean }?,

  ## default value: true
  attribute showDrill { xsd:boolean }?,

  ## default value: false
  attribute printDrill { xsd:boolean }?,

  ## default value: true
  attribute showMemberPropertyTips { xsd:boolean }?,

  ## default value: true
  attribute showDataTips { xsd:boolean }?,

  ## default value: true
  attribute enableWizard { xsd:boolean }?,

  ## default value: true
  attribute enableDrill { xsd:boolean }?,
```

```
  ## default value: true
  attribute enableFieldProperties { xsd:boolean }?,

  ## default value: true
  attribute preserveFormatting { xsd:boolean }?,

  ## default value: false
  attribute useAutoFormatting { xsd:boolean }?,

  ## default value: 0
  attribute pageWrap { xsd:unsignedInt }?,

  ## default value: false
  attribute pageOverThenDown { xsd:boolean }?,

  ## default value: false
  attribute subtotalHiddenItems { xsd:boolean }?,

  ## default value: true
  attribute rowGrandTotals { xsd:boolean }?,

  ## default value: true
  attribute colGrandTotals { xsd:boolean }?,

  ## default value: false
  attribute fieldPrintTitles { xsd:boolean }?,

  ## default value: false
  attribute itemPrintTitles { xsd:boolean }?,

  ## default value: false
  attribute mergeItem { xsd:boolean }?,

  ## default value: true
  attribute showDropZones { xsd:boolean }?,

  ## default value: 0
  attribute createdVersion { xsd:unsignedByte }?,

  ## default value: 1
  attribute indent { xsd:unsignedInt }?,

  ## default value: false
  attribute showEmptyRow { xsd:boolean }?,

  ## default value: false
  attribute showEmptyCol { xsd:boolean }?,

  ## default value: true
  attribute showHeaders { xsd:boolean }?,

  ## default value: true
```



```
  attribute compact { xsd:boolean }?,

  ## default value: false
  attribute outline { xsd:boolean }?,

  ## default value: false
  attribute outlineData { xsd:boolean }?,

  ## default value: true
  attribute compactData { xsd:boolean }?,

  ## default value: false
  attribute published { xsd:boolean }?,

  ## default value: false
  attribute gridDropZones { xsd:boolean }?,

  ## default value: true
  attribute immersive { xsd:boolean }?,

  ## default value: true
  attribute multipleFieldFilters { xsd:boolean }?,

  ## default value: 0
  attribute chartFormat { xsd:unsignedInt }?,
  attribute rowHeaderCaption { s_ST_Xstring }?,
  attribute colHeaderCaption { s_ST_Xstring }?,

  ## default value: false
  attribute fieldListSortAscending { xsd:boolean }?,

  ## default value: false
  attribute mdxSubqueries { xsd:boolean }?,

  ## default value: true
  attribute customListSort { xsd:boolean }?,
  element location { sml_CT_Location },
  element pivotFields { sml_CT_PivotFields }?,
  element rowFields { sml_CT_RowFields }?,
  element rowItems { sml_CT_rowItems }?,
  element colFields { sml_CT_ColFields }?,
  element colItems { sml_CT_colItems }?,
  element pageFields { sml_CT_PageFields }?,
  element dataFields { sml_CT_DataFields }?,
  element formats { sml_CT_Formats }?,
  element conditionalFormats { sml_CT_ConditionalFormats }?,
  element chartFormats { sml_CT_ChartFormats }?,
  element pivotHierarchies { sml_CT_PivotHierarchies }?,
  element pivotTableStyleInfo { sml_CT_PivotTableStyle }?,
  element filters { sml_CT_PivotFilters }?,
  element rowHierarchiesUsage { sml_CT_RowHierarchiesUsage }?,
  element colHierarchiesUsage { sml_CT_ColHierarchiesUsage }?,
  element extLst { sml_CT_ExtensionList }?
```

```
sml_CT_Location =
  attribute ref { sml_ST_Ref },
  attribute firstHeaderRow { xsd:unsignedInt },
  attribute firstDataRow { xsd:unsignedInt },
  attribute firstDataCol { xsd:unsignedInt },

  ## default value: 0
  attribute rowPageCount { xsd:unsignedInt }?,

  ## default value: 0
  attribute colPageCount { xsd:unsignedInt }?
sml_CT_PivotFields =
  attribute count { xsd:unsignedInt }?,
  element pivotField { sml_CT_PivotField }+
sml_CT_PivotField =
  attribute name { s_ST_Xstring }?,
  attribute axis { sml_ST_Axis }?,

  ## default value: false
  attribute dataField { xsd:boolean }?,
  attribute subtotalCaption { s_ST_Xstring }?,

  ## default value: true
  attribute showDropDowns { xsd:boolean }?,

  ## default value: false
  attribute hiddenLevel { xsd:boolean }?,
  attribute uniqueMemberProperty { s_ST_Xstring }?,

  ## default value: true
  attribute compact { xsd:boolean }?,

  ## default value: false
  attribute allDrilled { xsd:boolean }?,
  attribute numFmtId { sml_ST_NumFmtId }?,

  ## default value: true
  attribute outline { xsd:boolean }?,

  ## default value: true
  attribute subtotalTop { xsd:boolean }?,

  ## default value: true
  attribute dragToRow { xsd:boolean }?,

  ## default value: true
  attribute dragToCol { xsd:boolean }?,

  ## default value: false
  attribute multipleItemSelectionAllowed { xsd:boolean }?,

  ## default value: true
  attribute dragToPage { xsd:boolean }?,
```



```
  ## default value: true
  attribute dragToData { xsd:boolean }?,

  ## default value: true
  attribute dragOff { xsd:boolean }?,

  ## default value: true
  attribute showAll { xsd:boolean }?,

  ## default value: false
  attribute insertBlankRow { xsd:boolean }?,

  ## default value: false
  attribute serverField { xsd:boolean }?,

  ## default value: false
  attribute insertPageBreak { xsd:boolean }?,

  ## default value: false
  attribute autoShow { xsd:boolean }?,

  ## default value: true
  attribute topAutoShow { xsd:boolean }?,

  ## default value: false
  attribute hideNewItems { xsd:boolean }?,

  ## default value: false
  attribute measureFilter { xsd:boolean }?,

  ## default value: false
  attribute includeNewItemsInFilter { xsd:boolean }?,

  ## default value: 10
  attribute itemPageCount { xsd:unsignedInt }?,

  ## default value: manual
  attribute sortType { sml_ST_FieldSortType }?,
  attribute dataSourceSort { xsd:boolean }?,

  ## default value: false
  attribute nonAutoSortDefault { xsd:boolean }?,
  attribute rankBy { xsd:unsignedInt }?,

  ## default value: true
  attribute defaultSubtotal { xsd:boolean }?,

  ## default value: false
  attribute sumSubtotal { xsd:boolean }?,

  ## default value: false
  attribute countASubtotal { xsd:boolean }?,
```

```
  ## default value: false
  attribute avgSubtotal { xsd:boolean }?,

  ## default value: false
  attribute maxSubtotal { xsd:boolean }?,

  ## default value: false
  attribute minSubtotal { xsd:boolean }?,

  ## default value: false
  attribute productSubtotal { xsd:boolean }?,

  ## default value: false
  attribute countSubtotal { xsd:boolean }?,

  ## default value: false
  attribute stdDevSubtotal { xsd:boolean }?,

  ## default value: false
  attribute stdDevPSubtotal { xsd:boolean }?,

  ## default value: false
  attribute varSubtotal { xsd:boolean }?,

  ## default value: false
  attribute varPSubtotal { xsd:boolean }?,

  ## default value: false
  attribute showPropCell { xsd:boolean }?,

  ## default value: false
  attribute showPropTip { xsd:boolean }?,

  ## default value: false
  attribute showPropAsCaption { xsd:boolean }?,

  ## default value: false
  attribute defaultAttributeDrillState { xsd:boolean }?,
  element items { sml_CT_Items }?,
  element autoSortScope { sml_CT_AutoSortScope }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_AutoSortScope = element pivotArea { sml_CT_PivotArea }
sml_CT_Items =
  attribute count { xsd:unsignedInt }?,
  element item { sml_CT_Item }+
sml_CT_Item =
  attribute n { s_ST_Xstring }?,

  ## default value: data
  attribute t { sml_ST_ItemType }?,

  ## default value: false
```



```
  attribute h { xsd:boolean }?,

  ## default value: false
  attribute s { xsd:boolean }?,

  ## default value: true
  attribute sd { xsd:boolean }?,

  ## default value: false
  attribute f { xsd:boolean }?,

  ## default value: false
  attribute m { xsd:boolean }?,

  ## default value: false
  attribute c { xsd:boolean }?,
  attribute x { xsd:unsignedInt }?,

  ## default value: false
  attribute d { xsd:boolean }?,

  ## default value: true
  attribute e { xsd:boolean }?
sml_CT_PageFields =
  attribute count { xsd:unsignedInt }?,
  element pageField { sml_CT_PageField }+
sml_CT_PageField =
  attribute fld { xsd:int },
  attribute item { xsd:unsignedInt }?,
  attribute hier { xsd:int }?,
  attribute name { s_ST_Xstring }?,
  attribute cap { s_ST_Xstring }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_DataFields =
  attribute count { xsd:unsignedInt }?,
  element dataField { sml_CT_DataField }+
sml_CT_DataField =
  attribute name { s_ST_Xstring }?,
  attribute fld { xsd:unsignedInt },

  ## default value: sum
  attribute subtotal { sml_ST_DataConsolidateFunction }?,

  ## default value: normal
  attribute showDataAs { sml_ST_ShowDataAs }?,

  ## default value: -1
  attribute baseField { xsd:int }?,

  ## default value: 1048832
  attribute baseItem { xsd:unsignedInt }?,
  attribute numFmtId { sml_ST_NumFmtId }?,
  element extLst { sml_CT_ExtensionList }?
```

```
sml_CT_rowItems =
  attribute count { xsd:unsignedInt }?,
  element i { sml_CT_I }+
sml_CT_colItems =
  attribute count { xsd:unsignedInt }?,
  element i { sml_CT_I }+
sml_CT_I =

  ## default value: data
  attribute t { sml_ST_ItemType }?,

  ## default value: 0
  attribute r { xsd:unsignedInt }?,

  ## default value: 0
  attribute i { xsd:unsignedInt }?,
  element x { sml_CT_X }*
sml_CT_X =

  ## default value: 0
  attribute v { xsd:int }?
sml_CT_RowFields =

  ## default value: 0
  attribute count { xsd:unsignedInt }?,
  element field { sml_CT_Field }+
sml_CT_ColFields =

  ## default value: 0
  attribute count { xsd:unsignedInt }?,
  element field { sml_CT_Field }+
sml_CT_Field = attribute x { xsd:int }
sml_CT_Formats =

  ## default value: 0
  attribute count { xsd:unsignedInt }?,
  element format { sml_CT_Format }+
sml_CT_Format =

  ## default value: formatting
  attribute action { sml_ST_FormatAction }?,
  attribute dxfId { sml_ST_DxfId }?,
  element pivotArea { sml_CT_PivotArea },
  element extLst { sml_CT_ExtensionList }?
sml_CT_ConditionalFormats =

  ## default value: 0
  attribute count { xsd:unsignedInt }?,
  element conditionalFormat { sml_CT_ConditionalFormat }+
sml_CT_ConditionalFormat =

  ## default value: selection
  attribute scope { sml_ST_Scope }?,
```



```
  ## default value: none
  attribute type { sml_ST_Type }?,
  attribute priority { xsd:unsignedInt },
  element pivotAreas { sml_CT_PivotAreas },
  element extLst { sml_CT_ExtensionList }?
sml_CT_PivotAreas =
  attribute count { xsd:unsignedInt }?,
  element pivotArea { sml_CT_PivotArea }*
sml_ST_Scope = string "selection" | string "data" | string "field"
sml_ST_Type =
  string "none" | string "all" | string "row" | string "column"
sml_CT_ChartFormats =

  ## default value: 0
  attribute count { xsd:unsignedInt }?,
  element chartFormat { sml_CT_ChartFormat }+
sml_CT_ChartFormat =
  attribute chart { xsd:unsignedInt },
  attribute format { xsd:unsignedInt },

  ## default value: false
  attribute series { xsd:boolean }?,
  element pivotArea { sml_CT_PivotArea }
sml_CT_PivotHierarchies =
  attribute count { xsd:unsignedInt }?,
  element pivotHierarchy { sml_CT_PivotHierarchy }+
sml_CT_PivotHierarchy =

  ## default value: false
  attribute outline { xsd:boolean }?,

  ## default value: false
  attribute multipleItemSelectionAllowed { xsd:boolean }?,

  ## default value: false
  attribute subtotalTop { xsd:boolean }?,

  ## default value: true
  attribute showInFieldList { xsd:boolean }?,

  ## default value: true
  attribute dragToRow { xsd:boolean }?,

  ## default value: true
  attribute dragToCol { xsd:boolean }?,

  ## default value: true
  attribute dragToPage { xsd:boolean }?,

  ## default value: false
  attribute dragToData { xsd:boolean }?,

```

```
  ## default value: true
  attribute dragOff { xsd:boolean }?,

  ## default value: false
  attribute includeNewItemsInFilter { xsd:boolean }?,
  attribute caption { s_ST_Xstring }?,
  element mps { sml_CT_MemberProperties }?,
  element members { sml_CT_Members }*,
  element extLst { sml_CT_ExtensionList }?
sml_CT_RowHierarchiesUsage =
  attribute count { xsd:unsignedInt }?,
  element rowHierarchyUsage { sml_CT_HierarchyUsage }+
sml_CT_ColHierarchiesUsage =
  attribute count { xsd:unsignedInt }?,
  element colHierarchyUsage { sml_CT_HierarchyUsage }+
sml_CT_HierarchyUsage = attribute hierarchyUsage { xsd:int }
sml_CT_MemberProperties =
  attribute count { xsd:unsignedInt }?,
  element mp { sml_CT_MemberProperty }+
sml_CT_MemberProperty =
  attribute name { s_ST_Xstring }?,

  ## default value: false
  attribute showCell { xsd:boolean }?,

  ## default value: false
  attribute showTip { xsd:boolean }?,

  ## default value: false
  attribute showAsCaption { xsd:boolean }?,
  attribute nameLen { xsd:unsignedInt }?,
  attribute pPos { xsd:unsignedInt }?,
  attribute pLen { xsd:unsignedInt }?,
  attribute level { xsd:unsignedInt }?,
  attribute field { xsd:unsignedInt }
sml_CT_Members =
  attribute count { xsd:unsignedInt }?,
  attribute level { xsd:unsignedInt }?,
  element member { sml_CT_Member }+
sml_CT_Member = attribute name { s_ST_Xstring }
sml_CT_Dimensions =
  attribute count { xsd:unsignedInt }?,
  element dimension { sml_CT_PivotDimension }*
sml_CT_PivotDimension =

  ## default value: false
  attribute measure { xsd:boolean }?,
  attribute name { s_ST_Xstring },
  attribute uniqueName { s_ST_Xstring },
  attribute caption { s_ST_Xstring }
sml_CT_MeasureGroups =
  attribute count { xsd:unsignedInt }?,
  element measureGroup { sml_CT_MeasureGroup }*
```



```
sml_CT_MeasureDimensionMaps =
  attribute count { xsd:unsignedInt }?,
  element map { sml_CT_MeasureDimensionMap }*
sml_CT_MeasureGroup =
  attribute name { s_ST_Xstring },
  attribute caption { s_ST_Xstring }
sml_CT_MeasureDimensionMap =
  attribute measureGroup { xsd:unsignedInt }?,
  attribute dimension { xsd:unsignedInt }?
sml_CT_PivotTableStyle =
  attribute name { xsd:string }?,
  attribute showRowHeaders { xsd:boolean }?,
  attribute showColHeaders { xsd:boolean }?,
  attribute showRowStripes { xsd:boolean }?,
  attribute showColStripes { xsd:boolean }?,
  attribute showLastColumn { xsd:boolean }?
sml_CT_PivotFilters =

  ## default value: 0
  attribute count { xsd:unsignedInt }?,
  element filter { sml_CT_PivotFilter }*
sml_CT_PivotFilter =
  attribute fld { xsd:unsignedInt },
  attribute mpFld { xsd:unsignedInt }?,
  attribute type { sml_ST_PivotFilterType },

  ## default value: 0
  attribute evalOrder { xsd:int }?,
  attribute id { xsd:unsignedInt },
  attribute iMeasureHier { xsd:unsignedInt }?,
  attribute iMeasureFld { xsd:unsignedInt }?,
  attribute name { s_ST_Xstring }?,
  attribute description { s_ST_Xstring }?,
  attribute stringValue1 { s_ST_Xstring }?,
  attribute stringValue2 { s_ST_Xstring }?,
  element autoFilter { sml_CT_AutoFilter },
  element extLst { sml_CT_ExtensionList }?
sml_ST_ShowDataAs =
  string "normal"
  | string "difference"
  | string "percent"
  | string "percentDiff"
  | string "runTotal"
  | string "percentOfRow"
  | string "percentOfCol"
  | string "percentOfTotal"
  | string "index"
sml_ST_ItemType =
  string "data"
  | string "default"
  | string "sum"
  | string "countA"
  | string "avg"
```

```
  | string "max"
  | string "min"
  | string "product"
  | string "count"
  | string "stdDev"
  | string "stdDevP"
  | string "var"
  | string "varP"
  | string "grand"
  | string "blank"
sml_ST_FormatAction =
  string "blank"
  | string "formatting"
  | string "drill"
  | string "formula"
sml_ST_FieldSortType =
  string "manual" | string "ascending" | string "descending"
sml_ST_PivotFilterType =
  string "unknown"
  | string "count"
  | string "percent"
  | string "sum"
  | string "captionEqual"
  | string "captionNotEqual"
  | string "captionBeginsWith"
  | string "captionNotBeginsWith"
  | string "captionEndsWith"
  | string "captionNotEndsWith"
  | string "captionContains"
  | string "captionNotContains"
  | string "captionGreaterThan"
  | string "captionGreaterThanOrEqual"
  | string "captionLessThan"
  | string "captionLessThanOrEqual"
  | string "captionBetween"
  | string "captionNotBetween"
  | string "valueEqual"
  | string "valueNotEqual"
  | string "valueGreaterThan"
  | string "valueGreaterThanOrEqual"
  | string "valueLessThan"
  | string "valueLessThanOrEqual"
  | string "valueBetween"
  | string "valueNotBetween"
  | string "dateEqual"
  | string "dateNotEqual"
  | string "dateOlderThan"
  | string "dateOlderThanOrEqual"
  | string "dateNewerThan"
  | string "dateNewerThanOrEqual"
  | string "dateBetween"
  | string "dateNotBetween"
  | string "tomorrow"
```



```
    | string "today"
    | string "yesterday"
    | string "nextWeek"
    | string "thisWeek"
    | string "lastWeek"
    | string "nextMonth"
    | string "thisMonth"
    | string "lastMonth"
    | string "nextQuarter"
    | string "thisQuarter"
    | string "lastQuarter"
    | string "nextYear"
    | string "thisYear"
    | string "lastYear"
    | string "yearToDate"
    | string "Q1"
    | string "Q2"
    | string "Q3"
    | string "Q4"
    | string "M1"
    | string "M2"
    | string "M3"
    | string "M4"
    | string "M5"
    | string "M6"
    | string "M7"
    | string "M8"
    | string "M9"
    | string "M10"
    | string "M11"
    | string "M12"
sml_CT_PivotArea =
  attribute field { xsd:int }?,

  ## default value: normal
  attribute type { sml_ST_PivotAreaType }?,

  ## default value: true
  attribute dataOnly { xsd:boolean }?,

  ## default value: false
  attribute labelOnly { xsd:boolean }?,

  ## default value: false
  attribute grandRow { xsd:boolean }?,

  ## default value: false
  attribute grandCol { xsd:boolean }?,

  ## default value: false
  attribute cacheIndex { xsd:boolean }?,

  ## default value: true
```

```
  attribute outline { xsd:boolean }?,
  attribute offset { sml_ST_Ref }?,

  ## default value: false
  attribute collapsedLevelsAreSubtotals { xsd:boolean }?,
  attribute axis { sml_ST_Axis }?,
  attribute fieldPosition { xsd:unsignedInt }?,
  element references { sml_CT_PivotAreaReferences }?,
  element extLst { sml_CT_ExtensionList }?
sml_ST_PivotAreaType =
  string "none"
  | string "normal"
  | string "data"
  | string "all"
  | string "origin"
  | string "button"
  | string "topEnd"
  | string "topRight"
sml_CT_PivotAreaReferences =
  attribute count { xsd:unsignedInt }?,
  element reference { sml_CT_PivotAreaReference }+
sml_CT_PivotAreaReference =
  attribute field { xsd:unsignedInt }?,
  attribute count { xsd:unsignedInt }?,

  ## default value: true
  attribute selected { xsd:boolean }?,

  ## default value: false
  attribute byPosition { xsd:boolean }?,

  ## default value: false
  attribute relative { xsd:boolean }?,

  ## default value: false
  attribute defaultSubtotal { xsd:boolean }?,

  ## default value: false
  attribute sumSubtotal { xsd:boolean }?,

  ## default value: false
  attribute countASubtotal { xsd:boolean }?,

  ## default value: false
  attribute avgSubtotal { xsd:boolean }?,

  ## default value: false
  attribute maxSubtotal { xsd:boolean }?,

  ## default value: false
  attribute minSubtotal { xsd:boolean }?,

  ## default value: false
```



```
  attribute productSubtotal { xsd:boolean }?,

  ## default value: false
  attribute countSubtotal { xsd:boolean }?,

  ## default value: false
  attribute stdDevSubtotal { xsd:boolean }?,

  ## default value: false
  attribute stdDevPSubtotal { xsd:boolean }?,

  ## default value: false
  attribute varSubtotal { xsd:boolean }?,

  ## default value: false
  attribute varPSubtotal { xsd:boolean }?,
  element x { sml_CT_Index }*,
  element extLst { sml_CT_ExtensionList }?
sml_CT_Index = attribute v { xsd:unsignedInt }
sml_ST_Axis =
  string "axisRow"
  | string "axisCol"
  | string "axisPage"
  | string "axisValues"
sml_queryTable = element queryTable { sml_CT_QueryTable }
sml_CT_QueryTable =
  attribute name { s_ST_Xstring },

  ## default value: true
  attribute headers { xsd:boolean }?,

  ## default value: false
  attribute rowNumbers { xsd:boolean }?,

  ## default value: false
  attribute disableRefresh { xsd:boolean }?,

  ## default value: true
  attribute backgroundRefresh { xsd:boolean }?,

  ## default value: false
  attribute firstBackgroundRefresh { xsd:boolean }?,

  ## default value: false
  attribute refreshOnLoad { xsd:boolean }?,

  ## default value: insertDelete
  attribute growShrinkType { sml_ST_GrowShrinkType }?,

  ## default value: false
  attribute fillFormulas { xsd:boolean }?,

  ## default value: false
```

```
  attribute removeDataOnSave { xsd:boolean }?,

  ## default value: false
  attribute disableEdit { xsd:boolean }?,

  ## default value: true
  attribute preserveFormatting { xsd:boolean }?,

  ## default value: true
  attribute adjustColumnWidth { xsd:boolean }?,

  ## default value: false
  attribute intermediate { xsd:boolean }?,
  attribute connectionId { xsd:unsignedInt },
  sml_AG_AutoFormat,
  element queryTableRefresh { sml_CT_QueryTableRefresh }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_QueryTableRefresh =

  ## default value: true
  attribute preserveSortFilterLayout { xsd:boolean }?,

  ## default value: false
  attribute fieldIdWrapped { xsd:boolean }?,

  ## default value: true
  attribute headersInLastRefresh { xsd:boolean }?,

  ## default value: 0
  attribute minimumVersion { xsd:unsignedByte }?,

  ## default value: 1
  attribute nextId { xsd:unsignedInt }?,

  ## default value: 0
  attribute unboundColumnsLeft { xsd:unsignedInt }?,

  ## default value: 0
  attribute unboundColumnsRight { xsd:unsignedInt }?,
  element queryTableFields { sml_CT_QueryTableFields },
  element queryTableDeletedFields { sml_CT_QueryTableDeletedFields }?,
  element sortState { sml_CT_SortState }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_QueryTableDeletedFields =
  attribute count { xsd:unsignedInt }?,
  element deletedField { sml_CT_DeletedField }+
sml_CT_DeletedField = attribute name { s_ST_Xstring }
sml_CT_QueryTableFields =

  ## default value: 0
  attribute count { xsd:unsignedInt }?,
  element queryTableField { sml_CT_QueryTableField }*
sml_CT_QueryTableField =
```



```
  attribute id { xsd:unsignedInt },
  attribute name { s_ST_Xstring }?,

  ## default value: true
  attribute dataBound { xsd:boolean }?,

  ## default value: false
  attribute rowNumbers { xsd:boolean }?,

  ## default value: false
  attribute fillFormulas { xsd:boolean }?,

  ## default value: false
  attribute clipped { xsd:boolean }?,

  ## default value: 0
  attribute tableColumnId { xsd:unsignedInt }?,
  element extLst { sml_CT_ExtensionList }?
sml_ST_GrowShrinkType =
  string "insertDelete" | string "insertClear" | string "overwriteClear"
sml_sst = element sst { sml_CT_Sst }
sml_CT_Sst =
  attribute count { xsd:unsignedInt }?,
  attribute uniqueCount { xsd:unsignedInt }?,
  element si { sml_CT_Rst }*,
  element extLst { sml_CT_ExtensionList }?
sml_ST_PhoneticType =
  string "halfwidthKatakana"
  | string "fullwidthKatakana"
  | string "Hiragana"
  | string "noConversion"
sml_ST_PhoneticAlignment =
  string "noControl"
  | string "left"
  | string "center"
  | string "distributed"
sml_CT_PhoneticRun =
  attribute sb { xsd:unsignedInt },
  attribute eb { xsd:unsignedInt },
  element t { s_ST_Xstring }
sml_CT_RElt =
  element rPr { sml_CT_RPrElt }?,
  element t { s_ST_Xstring }
sml_CT_RPrElt =
  (element rFont { sml_CT_FontName }?
   | element charset { sml_CT_IntProperty }?
   | element family { sml_CT_IntProperty }?
   | element b { sml_CT_BooleanProperty }?
   | element i { sml_CT_BooleanProperty }?
   | element strike { sml_CT_BooleanProperty }?
   | element outline { sml_CT_BooleanProperty }?
   | element shadow { sml_CT_BooleanProperty }?
   | element condense { sml_CT_BooleanProperty }?
```

```
    | element extend { sml_CT_BooleanProperty }?
    | element color { sml_CT_Color }?
    | element sz { sml_CT_FontSize }?
    | element u { sml_CT_UnderlineProperty }?
    | element vertAlign { sml_CT_VerticalAlignFontProperty }?
    | element scheme { sml_CT_FontScheme }?)+
sml_CT_Rst =
  element t { s_ST_Xstring }?,
  element r { sml_CT_RElt }*,
  element rPh { sml_CT_PhoneticRun }*,
  element phoneticPr { sml_CT_PhoneticPr }?
sml_CT_PhoneticPr =
  attribute fontId { sml_ST_FontId },

  ## default value: fullwidthKatakana
  attribute type { sml_ST_PhoneticType }?,

  ## default value: left
  attribute alignment { sml_ST_PhoneticAlignment }?
sml_headers = element headers { sml_CT_RevisionHeaders }
sml_revisions = element revisions { sml_CT_Revisions }
sml_CT_RevisionHeaders =
  attribute guid { s_ST_Guid },
  attribute lastGuid { s_ST_Guid }?,

  ## default value: true
  attribute shared { xsd:boolean }?,

  ## default value: false
  attribute diskRevisions { xsd:boolean }?,

  ## default value: true
  attribute history { xsd:boolean }?,

  ## default value: true
  attribute trackRevisions { xsd:boolean }?,

  ## default value: false
  attribute exclusive { xsd:boolean }?,

  ## default value: 0
  attribute revisionId { xsd:unsignedInt }?,

  ## default value: 1
  attribute version { xsd:int }?,

  ## default value: true
  attribute keepChangeHistory { xsd:boolean }?,

  ## default value: false
  attribute protected { xsd:boolean }?,

  ## default value: 30
```



```
  attribute preserveHistory { xsd:unsignedInt }?,
  element header { sml_CT_RevisionHeader }+
sml_CT_Revisions =
  (element rrc { sml_CT_RevisionRowColumn }*
   | element rm { sml_CT_RevisionMove }*
   | element rcv { sml_CT_RevisionCustomView }*
   | element rsnm { sml_CT_RevisionSheetRename }*
   | element ris { sml_CT_RevisionInsertSheet }*
   | element rcc { sml_CT_RevisionCellChange }*
   | element rfmt { sml_CT_RevisionFormatting }*
   | element raf { sml_CT_RevisionAutoFormatting }*
   | element rdn { sml_CT_RevisionDefinedName }*
   | element rcmt { sml_CT_RevisionComment }*
   | element rqt { sml_CT_RevisionQueryTableField }*
   | element rcft { sml_CT_RevisionConflict }*)+
sml_AG_RevData =
  attribute rId { xsd:unsignedInt },

  ## default value: false
  attribute ua { xsd:boolean }?,

  ## default value: false
  attribute ra { xsd:boolean }?
sml_CT_RevisionHeader =
  attribute guid { s_ST_Guid },
  attribute dateTime { xsd:dateTime },
  attribute maxSheetId { xsd:unsignedInt },
  attribute userName { s_ST_Xstring },
  r_id,
  attribute minRId { xsd:unsignedInt }?,
  attribute maxRId { xsd:unsignedInt }?,
  element sheetIdMap { sml_CT_SheetIdMap },
  element reviewedList { sml_CT_ReviewedRevisions }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_SheetIdMap =
  attribute count { xsd:unsignedInt }?,
  element sheetId { sml_CT_SheetId }+
sml_CT_SheetId = attribute val { xsd:unsignedInt }
sml_CT_ReviewedRevisions =
  attribute count { xsd:unsignedInt }?,
  element reviewed { sml_CT_Reviewed }+
sml_CT_Reviewed = attribute rId { xsd:unsignedInt }
sml_CT_UndoInfo =
  attribute index { xsd:unsignedInt },
  attribute exp { sml_ST_FormulaExpression },

  ## default value: false
  attribute ref3D { xsd:boolean }?,

  ## default value: false
  attribute array { xsd:boolean }?,

  ## default value: false
```

```
  attribute v { xsd:boolean }?,

  ## default value: false
  attribute nf { xsd:boolean }?,

  ## default value: false
  attribute cs { xsd:boolean }?,
  attribute dr { sml_ST_RefA },
  attribute dn { s_ST_Xstring }?,
  attribute r { sml_ST_CellRef }?,
  attribute sId { xsd:unsignedInt }?
sml_CT_RevisionRowColumn =
  sml_AG_RevData,
  attribute sId { xsd:unsignedInt },

  ## default value: false
  attribute eol { xsd:boolean }?,
  attribute ref { sml_ST_Ref },
  attribute action { sml_ST_rwColActionType },

  ## default value: false
  attribute edge { xsd:boolean }?,
  (element undo { sml_CT_UndoInfo }*
   | element rcc { sml_CT_RevisionCellChange }*
   | element rfmt { sml_CT_RevisionFormatting }*)*
sml_CT_RevisionMove =
  sml_AG_RevData,
  attribute sheetId { xsd:unsignedInt },
  attribute source { sml_ST_Ref },
  attribute destination { sml_ST_Ref },

  ## default value: 0
  attribute sourceSheetId { xsd:unsignedInt }?,
  (element undo { sml_CT_UndoInfo }*
   | element rcc { sml_CT_RevisionCellChange }*
   | element rfmt { sml_CT_RevisionFormatting }*)*
sml_CT_RevisionCustomView =
  attribute guid { s_ST_Guid },
  attribute action { sml_ST_RevisionAction }
sml_CT_RevisionSheetRename =
  sml_AG_RevData,
  attribute sheetId { xsd:unsignedInt },
  attribute oldName { s_ST_Xstring },
  attribute newName { s_ST_Xstring },
  element extLst { sml_CT_ExtensionList }?
sml_CT_RevisionInsertSheet =
  sml_AG_RevData,
  attribute sheetId { xsd:unsignedInt },
  attribute name { s_ST_Xstring },
  attribute sheetPosition { xsd:unsignedInt }
sml_CT_RevisionCellChange =
  sml_AG_RevData,
  attribute sId { xsd:unsignedInt },
```



```
  ## default value: false
  attribute odxf { xsd:boolean }?,

  ## default value: false
  attribute xfDxf { xsd:boolean }?,

  ## default value: false
  attribute s { xsd:boolean }?,

  ## default value: false
  attribute dxf { xsd:boolean }?,
  attribute numFmtId { sml_ST_NumFmtId }?,

  ## default value: false
  attribute quotePrefix { xsd:boolean }?,

  ## default value: false
  attribute oldQuotePrefix { xsd:boolean }?,

  ## default value: false
  attribute ph { xsd:boolean }?,

  ## default value: false
  attribute oldPh { xsd:boolean }?,

  ## default value: false
  attribute endOfListFormulaUpdate { xsd:boolean }?,
  element oc { sml_CT_Cell }?,
  element nc { sml_CT_Cell },
  element odxf { sml_CT_Dxf }?,
  element ndxf { sml_CT_Dxf }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_RevisionFormatting =
  attribute sheetId { xsd:unsignedInt },

  ## default value: false
  attribute xfDxf { xsd:boolean }?,

  ## default value: false
  attribute s { xsd:boolean }?,
  attribute sqref { sml_ST_Sqref },
  attribute start { xsd:unsignedInt }?,
  attribute length { xsd:unsignedInt }?,
  element dxf { sml_CT_Dxf }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_RevisionAutoFormatting =
  attribute sheetId { xsd:unsignedInt },
  sml_AG_AutoFormat,
  attribute ref { sml_ST_Ref }
sml_CT_RevisionComment =
  attribute sheetId { xsd:unsignedInt },
  attribute cell { sml_ST_CellRef },
```

```
  attribute guid { s_ST_Guid },

  ## default value: add
  attribute action { sml_ST_RevisionAction }?,

  ## default value: false
  attribute alwaysShow { xsd:boolean }?,

  ## default value: false
  attribute old { xsd:boolean }?,

  ## default value: false
  attribute hiddenRow { xsd:boolean }?,

  ## default value: false
  attribute hiddenColumn { xsd:boolean }?,
  attribute author { s_ST_Xstring },

  ## default value: 0
  attribute oldLength { xsd:unsignedInt }?,

  ## default value: 0
  attribute newLength { xsd:unsignedInt }?
sml_CT_RevisionDefinedName =
  sml_AG_RevData,
  attribute localSheetId { xsd:unsignedInt }?,

  ## default value: false
  attribute customView { xsd:boolean }?,
  attribute name { s_ST_Xstring },

  ## default value: false
  attribute function { xsd:boolean }?,

  ## default value: false
  attribute oldFunction { xsd:boolean }?,
  attribute functionGroupId { xsd:unsignedByte }?,
  attribute oldFunctionGroupId { xsd:unsignedByte }?,
  attribute shortcutKey { xsd:unsignedByte }?,
  attribute oldShortcutKey { xsd:unsignedByte }?,

  ## default value: false
  attribute hidden { xsd:boolean }?,

  ## default value: false
  attribute oldHidden { xsd:boolean }?,
  attribute customMenu { s_ST_Xstring }?,
  attribute oldCustomMenu { s_ST_Xstring }?,
  attribute description { s_ST_Xstring }?,
  attribute oldDescription { s_ST_Xstring }?,
  attribute help { s_ST_Xstring }?,
  attribute oldHelp { s_ST_Xstring }?,
  attribute statusBar { s_ST_Xstring }?,
```



```
  attribute oldStatusBar { s_ST_Xstring }?,
  attribute comment { s_ST_Xstring }?,
  attribute oldComment { s_ST_Xstring }?,
  element formula { sml_ST_Formula }?,
  element oldFormula { sml_ST_Formula }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_RevisionConflict =
  sml_AG_RevData,
  attribute sheetId { xsd:unsignedInt }?
sml_CT_RevisionQueryTableField =
  attribute sheetId { xsd:unsignedInt },
  attribute ref { sml_ST_Ref },
  attribute fieldId { xsd:unsignedInt }
sml_ST_rwColActionType =
  string "insertRow"
  | string "deleteRow"
  | string "insertCol"
  | string "deleteCol"
sml_ST_RevisionAction = string "add" | string "delete"
sml_ST_FormulaExpression =
  string "ref"
  | string "refError"
  | string "area"
  | string "areaError"
  | string "computedArea"
sml_users = element users { sml_CT_Users }
sml_CT_Users =
  attribute count { xsd:unsignedInt }?,
  element userInfo { sml_CT_SharedUser }*
sml_CT_SharedUser =
  attribute guid { s_ST_Guid },
  attribute name { s_ST_Xstring },
  attribute id { xsd:int },
  attribute dateTime { xsd:dateTime },
  element extLst { sml_CT_ExtensionList }?
sml_worksheet = element worksheet { sml_CT_Worksheet }
sml_chartsheet = element chartsheet { sml_CT_Chartsheet }
sml_dialogsheet = element dialogsheet { sml_CT_Dialogsheet }
sml_CT_Macrosheet =
  element sheetPr { sml_CT_SheetPr }?,
  element dimension { sml_CT_SheetDimension }?,
  element sheetViews { sml_CT_SheetViews }?,
  element sheetFormatPr { sml_CT_SheetFormatPr }?,
  element cols { sml_CT_Cols }*,
  element sheetData { sml_CT_SheetData },
  element sheetProtection { sml_CT_SheetProtection }?,
  element autoFilter { sml_CT_AutoFilter }?,
  element sortState { sml_CT_SortState }?,
  element dataConsolidate { sml_CT_DataConsolidate }?,
  element customSheetViews { sml_CT_CustomSheetViews }?,
  element phoneticPr { sml_CT_PhoneticPr }?,
  element conditionalFormatting { sml_CT_ConditionalFormatting }*,
  element printOptions { sml_CT_PrintOptions }?,
```

```
  element pageMargins { sml_CT_PageMargins }?,
  element pageSetup { sml_CT_PageSetup }?,
  element headerFooter { sml_CT_HeaderFooter }?,
  element rowBreaks { sml_CT_PageBreak }?,
  element colBreaks { sml_CT_PageBreak }?,
  element customProperties { sml_CT_CustomProperties }?,
  element drawing { sml_CT_Drawing }?,
  element legacyDrawing { sml_CT_LegacyDrawing }?,
  element legacyDrawingHF { sml_CT_LegacyDrawing }?,
  element drawingHF { sml_CT_DrawingHF }?,
  element picture { sml_CT_SheetBackgroundPicture }?,
  element oleObjects { sml_CT_OleObjects }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_Dialogsheet =
  element sheetPr { sml_CT_SheetPr }?,
  element sheetViews { sml_CT_SheetViews }?,
  element sheetFormatPr { sml_CT_SheetFormatPr }?,
  element sheetProtection { sml_CT_SheetProtection }?,
  element customSheetViews { sml_CT_CustomSheetViews }?,
  element printOptions { sml_CT_PrintOptions }?,
  element pageMargins { sml_CT_PageMargins }?,
  element pageSetup { sml_CT_PageSetup }?,
  element headerFooter { sml_CT_HeaderFooter }?,
  element drawing { sml_CT_Drawing }?,
  element legacyDrawing { sml_CT_LegacyDrawing }?,
  element legacyDrawingHF { sml_CT_LegacyDrawing }?,
  element drawingHF { sml_CT_DrawingHF }?,
  element oleObjects { sml_CT_OleObjects }?,
  element controls { sml_CT_Controls }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_Worksheet =
  element sheetPr { sml_CT_SheetPr }?,
  element dimension { sml_CT_SheetDimension }?,
  element sheetViews { sml_CT_SheetViews }?,
  element sheetFormatPr { sml_CT_SheetFormatPr }?,
  element cols { sml_CT_Cols }*,
  element sheetData { sml_CT_SheetData },
  element sheetCalcPr { sml_CT_SheetCalcPr }?,
  element sheetProtection { sml_CT_SheetProtection }?,
  element protectedRanges { sml_CT_ProtectedRanges }?,
  element scenarios { sml_CT_Scenarios }?,
  element autoFilter { sml_CT_AutoFilter }?,
  element sortState { sml_CT_SortState }?,
  element dataConsolidate { sml_CT_DataConsolidate }?,
  element customSheetViews { sml_CT_CustomSheetViews }?,
  element mergeCells { sml_CT_MergeCells }?,
  element phoneticPr { sml_CT_PhoneticPr }?,
  element conditionalFormatting { sml_CT_ConditionalFormatting }*,
  element dataValidations { sml_CT_DataValidations }?,
  element hyperlinks { sml_CT_Hyperlinks }?,
  element printOptions { sml_CT_PrintOptions }?,
  element pageMargins { sml_CT_PageMargins }?,
  element pageSetup { sml_CT_PageSetup }?,
```



```
  element headerFooter { sml_CT_HeaderFooter }?,
  element rowBreaks { sml_CT_PageBreak }?,
  element colBreaks { sml_CT_PageBreak }?,
  element customProperties { sml_CT_CustomProperties }?,
  element cellWatches { sml_CT_CellWatches }?,
  element ignoredErrors { sml_CT_IgnoredErrors }?,
  element smartTags { sml_CT_SmartTags }?,
  element drawing { sml_CT_Drawing }?,
  element legacyDrawing { sml_CT_LegacyDrawing }?,
  element legacyDrawingHF { sml_CT_LegacyDrawing }?,
  element drawingHF { sml_CT_DrawingHF }?,
  element picture { sml_CT_SheetBackgroundPicture }?,
  element oleObjects { sml_CT_OleObjects }?,
  element controls { sml_CT_Controls }?,
  element webPublishItems { sml_CT_WebPublishItems }?,
  element tableParts { sml_CT_TableParts }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_SheetData = element row { sml_CT_Row }*
sml_CT_SheetCalcPr =

  ## default value: false
  attribute fullCalcOnLoad { xsd:boolean }?
sml_CT_SheetFormatPr =

  ## default value: 8
  attribute baseColWidth { xsd:unsignedInt }?,
  attribute defaultColWidth { xsd:double }?,
  attribute defaultRowHeight { xsd:double },

  ## default value: false
  attribute customHeight { xsd:boolean }?,

  ## default value: false
  attribute zeroHeight { xsd:boolean }?,

  ## default value: false
  attribute thickTop { xsd:boolean }?,

  ## default value: false
  attribute thickBottom { xsd:boolean }?,

  ## default value: 0
  attribute outlineLevelRow { xsd:unsignedByte }?,

  ## default value: 0
  attribute outlineLevelCol { xsd:unsignedByte }?
sml_CT_Cols = element col { sml_CT_Col }+
sml_CT_Col =
  attribute min { xsd:unsignedInt },
  attribute max { xsd:unsignedInt },
  attribute width { xsd:double }?,

  ## default value: 0
```

```
  attribute style { xsd:unsignedInt }?,

  ## default value: false
  attribute hidden { xsd:boolean }?,

  ## default value: false
  attribute bestFit { xsd:boolean }?,

  ## default value: false
  attribute customWidth { xsd:boolean }?,

  ## default value: false
  attribute phonetic { xsd:boolean }?,

  ## default value: 0
  attribute outlineLevel { xsd:unsignedByte }?,

  ## default value: false
  attribute collapsed { xsd:boolean }?
sml_ST_CellSpan = xsd:string
sml_ST_CellSpans = list { sml_ST_CellSpan* }
sml_CT_Row =
  attribute r { xsd:unsignedInt }?,
  attribute spans { sml_ST_CellSpans }?,

  ## default value: 0
  attribute s { xsd:unsignedInt }?,

  ## default value: false
  attribute customFormat { xsd:boolean }?,
  attribute ht { xsd:double }?,

  ## default value: false
  attribute hidden { xsd:boolean }?,

  ## default value: false
  attribute customHeight { xsd:boolean }?,

  ## default value: 0
  attribute outlineLevel { xsd:unsignedByte }?,

  ## default value: false
  attribute collapsed { xsd:boolean }?,

  ## default value: false
  attribute thickTop { xsd:boolean }?,

  ## default value: false
  attribute thickBot { xsd:boolean }?,

  ## default value: false
  attribute ph { xsd:boolean }?,
  element c { sml_CT_Cell }*,
```



```
  element extLst { sml_CT_ExtensionList }?
sml_CT_Cell =
  attribute r { sml_ST_CellRef }?,

  ## default value: 0
  attribute s { xsd:unsignedInt }?,

  ## default value: n
  attribute t { sml_ST_CellType }?,

  ## default value: 0
  attribute cm { xsd:unsignedInt }?,

  ## default value: 0
  attribute vm { xsd:unsignedInt }?,

  ## default value: false
  attribute ph { xsd:boolean }?,
  element f { sml_CT_CellFormula }?,
  element v { s_ST_Xstring }?,
  element is { sml_CT_Rst }?,
  element extLst { sml_CT_ExtensionList }?
sml_ST_CellType =
  string "b"
  | string "n"
  | string "e"
  | string "s"
  | string "str"
  | string "inlineStr"
sml_ST_CellFormulaType =
  string "normal"
  | string "array"
  | string "dataTable"
  | string "shared"
sml_CT_SheetPr =

  ## default value: false
  attribute syncHorizontal { xsd:boolean }?,

  ## default value: false
  attribute syncVertical { xsd:boolean }?,
  attribute syncRef { sml_ST_Ref }?,

  ## default value: false
  attribute transitionEvaluation { xsd:boolean }?,

  ## default value: false
  attribute transitionEntry { xsd:boolean }?,

  ## default value: true
  attribute published { xsd:boolean }?,
  attribute codeName { xsd:string }?,

```

```
  ## default value: false
  attribute filterMode { xsd:boolean }?,

  ## default value: true
  attribute enableFormatConditionsCalculation { xsd:boolean }?,
  element tabColor { sml_CT_Color }?,
  element outlinePr { sml_CT_OutlinePr }?,
  element pageSetUpPr { sml_CT_PageSetUpPr }?
sml_CT_SheetDimension = attribute ref { sml_ST_Ref }
sml_CT_SheetViews =
  element sheetView { sml_CT_SheetView }+,
  element extLst { sml_CT_ExtensionList }?
sml_CT_SheetView =

  ## default value: false
  attribute windowProtection { xsd:boolean }?,

  ## default value: false
  attribute showFormulas { xsd:boolean }?,

  ## default value: true
  attribute showGridLines { xsd:boolean }?,

  ## default value: true
  attribute showRowColHeaders { xsd:boolean }?,

  ## default value: true
  attribute showZeros { xsd:boolean }?,

  ## default value: false
  attribute rightToLeft { xsd:boolean }?,

  ## default value: false
  attribute tabSelected { xsd:boolean }?,

  ## default value: true
  attribute showRuler { xsd:boolean }?,

  ## default value: true
  attribute showOutlineSymbols { xsd:boolean }?,

  ## default value: true
  attribute defaultGridColor { xsd:boolean }?,

  ## default value: true
  attribute showWhiteSpace { xsd:boolean }?,

  ## default value: normal
  attribute view { sml_ST_SheetViewType }?,
  attribute topLeftCell { sml_ST_CellRef }?,

  ## default value: 64
  attribute colorId { xsd:unsignedInt }?,
```



```

  ## default value: 100
  attribute zoomScale { xsd:unsignedInt }?,

  ## default value: 0
  attribute zoomScaleNormal { xsd:unsignedInt }?,

  ## default value: 0
  attribute zoomScaleSheetLayoutView { xsd:unsignedInt }?,

  ## default value: 0
  attribute zoomScalePageLayoutView { xsd:unsignedInt }?,
  attribute workbookViewId { xsd:unsignedInt },
  element pane { sml_CT_Pane }?,
  element selection { sml_CT_Selection }*,
  element pivotSelection { sml_CT_PivotSelection }*,
  element extLst { sml_CT_ExtensionList }?
sml_CT_Pane =

  ## default value: 0
  attribute xSplit { xsd:double }?,

  ## default value: 0
  attribute ySplit { xsd:double }?,
  attribute topLeftCell { sml_ST_CellRef }?,

  ## default value: topLeft
  attribute activePane { sml_ST_Pane }?,

  ## default value: split
  attribute state { sml_ST_PaneState }?
sml_CT_PivotSelection =

  ## default value: topLeft
  attribute pane { sml_ST_Pane }?,

  ## default value: false
  attribute showHeader { xsd:boolean }?,

  ## default value: false
  attribute label { xsd:boolean }?,

  ## default value: false
  attribute data { xsd:boolean }?,

  ## default value: false
  attribute extendable { xsd:boolean }?,

  ## default value: 0
  attribute count { xsd:unsignedInt }?,
  attribute axis { sml_ST_Axis }?,

  ## default value: 0
```

```
  attribute dimension { xsd:unsignedInt }?,

  ## default value: 0
  attribute start { xsd:unsignedInt }?,

  ## default value: 0
  attribute min { xsd:unsignedInt }?,

  ## default value: 0
  attribute max { xsd:unsignedInt }?,

  ## default value: 0
  attribute activeRow { xsd:unsignedInt }?,

  ## default value: 0
  attribute activeCol { xsd:unsignedInt }?,

  ## default value: 0
  attribute previousRow { xsd:unsignedInt }?,

  ## default value: 0
  attribute previousCol { xsd:unsignedInt }?,

  ## default value: 0
  attribute click { xsd:unsignedInt }?,
  r_id?,
  element pivotArea { sml_CT_PivotArea }
sml_CT_Selection =

  ## default value: topLeft
  attribute pane { sml_ST_Pane }?,
  attribute activeCell { sml_ST_CellRef }?,

  ## default value: 0
  attribute activeCellId { xsd:unsignedInt }?,

  ## default value: A1
  attribute sqref { sml_ST_Sqref }?
sml_ST_Pane =
  string "bottomRight"
  | string "topRight"
  | string "bottomLeft"
  | string "topLeft"
sml_CT_PageBreak =

  ## default value: 0
  attribute count { xsd:unsignedInt }?,

  ## default value: 0
  attribute manualBreakCount { xsd:unsignedInt }?,
  element brk { sml_CT_Break }*
sml_CT_Break =

```



```
  ## default value: 0
  attribute id { xsd:unsignedInt }?,

  ## default value: 0
  attribute min { xsd:unsignedInt }?,

  ## default value: 0
  attribute max { xsd:unsignedInt }?,

  ## default value: false
  attribute man { xsd:boolean }?,

  ## default value: false
  attribute pt { xsd:boolean }?
sml_ST_SheetViewType =
  string "normal" | string "pageBreakPreview" | string "pageLayout"
sml_CT_OutlinePr =

  ## default value: false
  attribute applyStyles { xsd:boolean }?,

  ## default value: true
  attribute summaryBelow { xsd:boolean }?,

  ## default value: true
  attribute summaryRight { xsd:boolean }?,

  ## default value: true
  attribute showOutlineSymbols { xsd:boolean }?
sml_CT_PageSetUpPr =

  ## default value: true
  attribute autoPageBreaks { xsd:boolean }?,

  ## default value: false
  attribute fitToPage { xsd:boolean }?
sml_CT_DataConsolidate =

  ## default value: sum
  attribute function { sml_ST_DataConsolidateFunction }?,

  ## default value: false
  attribute startLabels { xsd:boolean }?,

  ## default value: false
  attribute leftLabels { xsd:boolean }?,

  ## default value: false
  attribute topLabels { xsd:boolean }?,

  ## default value: false
  attribute link { xsd:boolean }?,
  element dataRefs { sml_CT_DataRefs }?
```

```
sml_ST_DataConsolidateFunction =
  string "average"
  | string "count"
  | string "countNums"
  | string "max"
  | string "min"
  | string "product"
  | string "stdDev"
  | string "stdDevp"
  | string "sum"
  | string "var"
  | string "varp"
sml_CT_DataRefs =
  attribute count { xsd:unsignedInt }?,
  element dataRef { sml_CT_DataRef }*
sml_CT_DataRef =
  attribute ref { sml_ST_Ref }?,
  attribute name { s_ST_Xstring }?,
  attribute sheet { s_ST_Xstring }?,
  r_id?
sml_CT_MergeCells =
  attribute count { xsd:unsignedInt }?,
  element mergeCell { sml_CT_MergeCell }+
sml_CT_MergeCell = attribute ref { sml_ST_Ref }
sml_CT_SmartTags = element cellSmartTags { sml_CT_CellSmartTags }+
sml_CT_CellSmartTags =
  attribute r { sml_ST_CellRef },
  element cellSmartTag { sml_CT_CellSmartTag }+
sml_CT_CellSmartTag =
  attribute type { xsd:unsignedInt },

  ## default value: false
  attribute deleted { xsd:boolean }?,

  ## default value: false
  attribute xmlBased { xsd:boolean }?,
  element cellSmartTagPr { sml_CT_CellSmartTagPr }*
sml_CT_CellSmartTagPr =
  attribute key { s_ST_Xstring },
  attribute val { s_ST_Xstring }
sml_CT_Drawing = r_id
sml_CT_LegacyDrawing = r_id
sml_CT_DrawingHF =
  r_id,
  attribute lho { xsd:unsignedInt }?,
  attribute lhe { xsd:unsignedInt }?,
  attribute lhf { xsd:unsignedInt }?,
  attribute cho { xsd:unsignedInt }?,
  attribute che { xsd:unsignedInt }?,
  attribute chf { xsd:unsignedInt }?,
  attribute rho { xsd:unsignedInt }?,
  attribute rhe { xsd:unsignedInt }?,
  attribute rhf { xsd:unsignedInt }?,
```



```
  attribute lfo { xsd:unsignedInt }?,
  attribute lfe { xsd:unsignedInt }?,
  attribute lff { xsd:unsignedInt }?,
  attribute cfo { xsd:unsignedInt }?,
  attribute cfe { xsd:unsignedInt }?,
  attribute cff { xsd:unsignedInt }?,
  attribute rfo { xsd:unsignedInt }?,
  attribute rfe { xsd:unsignedInt }?,
  attribute rff { xsd:unsignedInt }?
sml_CT_CustomSheetViews =
  element customSheetView { sml_CT_CustomSheetView }+
sml_CT_CustomSheetView =
  attribute guid { s_ST_Guid },

  ## default value: 100
  attribute scale { xsd:unsignedInt }?,

  ## default value: 64
  attribute colorId { xsd:unsignedInt }?,

  ## default value: false
  attribute showPageBreaks { xsd:boolean }?,

  ## default value: false
  attribute showFormulas { xsd:boolean }?,

  ## default value: true
  attribute showGridLines { xsd:boolean }?,

  ## default value: true
  attribute showRowCol { xsd:boolean }?,

  ## default value: true
  attribute outlineSymbols { xsd:boolean }?,

  ## default value: true
  attribute zeroValues { xsd:boolean }?,

  ## default value: false
  attribute fitToPage { xsd:boolean }?,

  ## default value: false
  attribute printArea { xsd:boolean }?,

  ## default value: false
  attribute filter { xsd:boolean }?,

  ## default value: false
  attribute showAutoFilter { xsd:boolean }?,

  ## default value: false
  attribute hiddenRows { xsd:boolean }?,

```

```
  ## default value: false
  attribute hiddenColumns { xsd:boolean }?,

  ## default value: visible
  attribute state { sml_ST_SheetState }?,

  ## default value: false
  attribute filterUnique { xsd:boolean }?,

  ## default value: normal
  attribute view { sml_ST_SheetViewType }?,

  ## default value: true
  attribute showRuler { xsd:boolean }?,
  attribute topLeftCell { sml_ST_CellRef }?,
  element pane { sml_CT_Pane }?,
  element selection { sml_CT_Selection }?,
  element rowBreaks { sml_CT_PageBreak }?,
  element colBreaks { sml_CT_PageBreak }?,
  element pageMargins { sml_CT_PageMargins }?,
  element printOptions { sml_CT_PrintOptions }?,
  element pageSetup { sml_CT_PageSetup }?,
  element headerFooter { sml_CT_HeaderFooter }?,
  element autoFilter { sml_CT_AutoFilter }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_DataValidations =

  ## default value: false
  attribute disablePrompts { xsd:boolean }?,
  attribute xWindow { xsd:unsignedInt }?,
  attribute yWindow { xsd:unsignedInt }?,
  attribute count { xsd:unsignedInt }?,
  element dataValidation { sml_CT_DataValidation }+
sml_CT_DataValidation =

  ## default value: none
  attribute type { sml_ST_DataValidationType }?,

  ## default value: stop
  attribute errorStyle { sml_ST_DataValidationErrorStyle }?,

  ## default value: noControl
  attribute imeMode { sml_ST_DataValidationImeMode }?,

  ## default value: between
  attribute operator { sml_ST_DataValidationOperator }?,

  ## default value: false
  attribute allowBlank { xsd:boolean }?,

  ## default value: false
  attribute showDropDown { xsd:boolean }?,

```



```
  ## default value: false
  attribute showInputMessage { xsd:boolean }?,

  ## default value: false
  attribute showErrorMessage { xsd:boolean }?,
  attribute errorTitle { s_ST_Xstring }?,
  attribute error { s_ST_Xstring }?,
  attribute promptTitle { s_ST_Xstring }?,
  attribute prompt { s_ST_Xstring }?,
  attribute sqref { sml_ST_Sqref },
  element formula1 { sml_ST_Formula }?,
  element formula2 { sml_ST_Formula }?
sml_ST_DataValidationType =
  string "none"
  | string "whole"
  | string "decimal"
  | string "list"
  | string "date"
  | string "time"
  | string "textLength"
  | string "custom"
sml_ST_DataValidationOperator =
  string "between"
  | string "notBetween"
  | string "equal"
  | string "notEqual"
  | string "lessThan"
  | string "lessThanOrEqual"
  | string "greaterThan"
  | string "greaterThanOrEqual"
sml_ST_DataValidationErrorStyle =
  string "stop" | string "warning" | string "information"
sml_ST_DataValidationImeMode =
  string "noControl"
  | string "off"
  | string "on"
  | string "disabled"
  | string "hiragana"
  | string "fullKatakana"
  | string "halfKatakana"
  | string "fullAlpha"
  | string "halfAlpha"
  | string "fullHangul"
  | string "halfHangul"
sml_ST_CfType =
  string "expression"
  | string "cellIs"
  | string "colorScale"
  | string "dataBar"
  | string "iconSet"
  | string "top10"
  | string "uniqueValues"
  | string "duplicateValues"
```

```
  | string "containsText"
  | string "notContainsText"
  | string "beginsWith"
  | string "endsWith"
  | string "containsBlanks"
  | string "notContainsBlanks"
  | string "containsErrors"
  | string "notContainsErrors"
  | string "timePeriod"
  | string "aboveAverage"
sml_ST_TimePeriod =
  string "today"
  | string "yesterday"
  | string "tomorrow"
  | string "last7Days"
  | string "thisMonth"
  | string "lastMonth"
  | string "nextMonth"
  | string "thisWeek"
  | string "lastWeek"
  | string "nextWeek"
sml_ST_ConditionalFormattingOperator =
  string "lessThan"
  | string "lessThanOrEqual"
  | string "equal"
  | string "notEqual"
  | string "greaterThanOrEqual"
  | string "greaterThan"
  | string "between"
  | string "notBetween"
  | string "containsText"
  | string "notContains"
  | string "beginsWith"
  | string "endsWith"
sml_ST_CfvoType =
  string "num"
  | string "percent"
  | string "max"
  | string "min"
  | string "formula"
  | string "percentile"
sml_CT_ConditionalFormatting =

  ## default value: false
  attribute pivot { xsd:boolean }?,
  attribute sqref { sml_ST_Sqref }?,
  element cfRule { sml_CT_CfRule }+,
  element extLst { sml_CT_ExtensionList }?
sml_CT_CfRule =
  attribute type { sml_ST_CfType }?,
  attribute dxfId { sml_ST_DxfId }?,
  attribute priority { xsd:int },

```



```
  ## default value: false
  attribute stopIfTrue { xsd:boolean }?,

  ## default value: true
  attribute aboveAverage { xsd:boolean }?,

  ## default value: false
  attribute percent { xsd:boolean }?,

  ## default value: false
  attribute bottom { xsd:boolean }?,
  attribute operator { sml_ST_ConditionalFormattingOperator }?,
  attribute text { xsd:string }?,
  attribute timePeriod { sml_ST_TimePeriod }?,
  attribute rank { xsd:unsignedInt }?,
  attribute stdDev { xsd:int }?,

  ## default value: false
  attribute equalAverage { xsd:boolean }?,
  element formula { sml_ST_Formula }*,
  element colorScale { sml_CT_ColorScale }?,
  element dataBar { sml_CT_DataBar }?,
  element iconSet { sml_CT_IconSet }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_Hyperlinks = element hyperlink { sml_CT_Hyperlink }+
sml_CT_Hyperlink =
  attribute ref { sml_ST_Ref },
  r_id?,
  attribute location { s_ST_Xstring }?,
  attribute tooltip { s_ST_Xstring }?,
  attribute display { s_ST_Xstring }?
sml_CT_CellFormula =
  sml_ST_Formula,

  ## default value: normal
  attribute t { sml_ST_CellFormulaType }?,

  ## default value: false
  attribute aca { xsd:boolean }?,
  attribute ref { sml_ST_Ref }?,

  ## default value: false
  attribute dt2D { xsd:boolean }?,

  ## default value: false
  attribute dtr { xsd:boolean }?,

  ## default value: false
  attribute del1 { xsd:boolean }?,

  ## default value: false
  attribute del2 { xsd:boolean }?,
  attribute r1 { sml_ST_CellRef }?,
```

```
  attribute r2 { sml_ST_CellRef }?,

  ## default value: false
  attribute ca { xsd:boolean }?,
  attribute si { xsd:unsignedInt }?,

  ## default value: false
  attribute bx { xsd:boolean }?
sml_CT_ColorScale =
  element cfvo { sml_CT_Cfvo }+,
  element color { sml_CT_Color }+
sml_CT_DataBar =

  ## default value: 10
  attribute minLength { xsd:unsignedInt }?,

  ## default value: 90
  attribute maxLength { xsd:unsignedInt }?,

  ## default value: true
  attribute showValue { xsd:boolean }?,
  element cfvo { sml_CT_Cfvo }+,
  element color { sml_CT_Color }
sml_CT_IconSet =

  ## default value: 3TrafficLights1
  attribute iconSet { sml_ST_IconSetType }?,

  ## default value: true
  attribute showValue { xsd:boolean }?,

  ## default value: true
  attribute percent { xsd:boolean }?,

  ## default value: false
  attribute reverse { xsd:boolean }?,
  element cfvo { sml_CT_Cfvo }+
sml_CT_Cfvo =
  attribute type { sml_ST_CfvoType },
  attribute val { s_ST_Xstring }?,

  ## default value: true
  attribute gte { xsd:boolean }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_PageMargins =
  attribute left { xsd:double },
  attribute right { xsd:double },
  attribute top { xsd:double },
  attribute bottom { xsd:double },
  attribute header { xsd:double },
  attribute footer { xsd:double }
sml_CT_PrintOptions =

```



```
  ## default value: false
  attribute horizontalCentered { xsd:boolean }?,

  ## default value: false
  attribute verticalCentered { xsd:boolean }?,

  ## default value: false
  attribute headings { xsd:boolean }?,

  ## default value: false
  attribute gridLines { xsd:boolean }?,

  ## default value: true
  attribute gridLinesSet { xsd:boolean }?
sml_CT_PageSetup =

  ## default value: 1
  attribute paperSize { xsd:unsignedInt }?,
  attribute paperHeight { s_ST_PositiveUniversalMeasure }?,
  attribute paperWidth { s_ST_PositiveUniversalMeasure }?,

  ## default value: 100
  attribute scale { xsd:unsignedInt }?,

  ## default value: 1
  attribute firstPageNumber { xsd:unsignedInt }?,

  ## default value: 1
  attribute fitToWidth { xsd:unsignedInt }?,

  ## default value: 1
  attribute fitToHeight { xsd:unsignedInt }?,

  ## default value: downThenOver
  attribute pageOrder { sml_ST_PageOrder }?,

  ## default value: default
  attribute orientation { sml_ST_Orientation }?,

  ## default value: true
  attribute usePrinterDefaults { xsd:boolean }?,

  ## default value: false
  attribute blackAndWhite { xsd:boolean }?,

  ## default value: false
  attribute draft { xsd:boolean }?,

  ## default value: none
  attribute cellComments { sml_ST_CellComments }?,

  ## default value: false
  attribute useFirstPageNumber { xsd:boolean }?,
```

```
  ## default value: displayed
  attribute errors { sml_ST_PrintError }?,

  ## default value: 600
  attribute horizontalDpi { xsd:unsignedInt }?,

  ## default value: 600
  attribute verticalDpi { xsd:unsignedInt }?,

  ## default value: 1
  attribute copies { xsd:unsignedInt }?,
  r_id?
sml_ST_PageOrder = string "downThenOver" | string "overThenDown"
sml_ST_Orientation =
  string "default" | string "portrait" | string "landscape"
sml_ST_CellComments =
  string "none" | string "asDisplayed" | string "atEnd"
sml_CT_HeaderFooter =

  ## default value: false
  attribute differentOddEven { xsd:boolean }?,

  ## default value: false
  attribute differentFirst { xsd:boolean }?,

  ## default value: true
  attribute scaleWithDoc { xsd:boolean }?,

  ## default value: true
  attribute alignWithMargins { xsd:boolean }?,
  element oddHeader { s_ST_Xstring }?,
  element oddFooter { s_ST_Xstring }?,
  element evenHeader { s_ST_Xstring }?,
  element evenFooter { s_ST_Xstring }?,
  element firstHeader { s_ST_Xstring }?,
  element firstFooter { s_ST_Xstring }?
sml_ST_PrintError =
  string "displayed" | string "blank" | string "dash" | string "NA"
sml_CT_Scenarios =
  attribute current { xsd:unsignedInt }?,
  attribute show { xsd:unsignedInt }?,
  attribute sqref { sml_ST_Sqref }?,
  element scenario { sml_CT_Scenario }+
sml_CT_SheetProtection =
  attribute password { sml_ST_UnsignedShortHex }?,
  attribute algorithmName { s_ST_Xstring }?,
  attribute hashValue { xsd:base64Binary }?,
  attribute saltValue { xsd:base64Binary }?,
  attribute spinCount { xsd:unsignedInt }?,

  ## default value: false
  attribute sheet { xsd:boolean }?,
```



```
  ## default value: false
  attribute objects { xsd:boolean }?,

  ## default value: false
  attribute scenarios { xsd:boolean }?,

  ## default value: true
  attribute formatCells { xsd:boolean }?,

  ## default value: true
  attribute formatColumns { xsd:boolean }?,

  ## default value: true
  attribute formatRows { xsd:boolean }?,

  ## default value: true
  attribute insertColumns { xsd:boolean }?,

  ## default value: true
  attribute insertRows { xsd:boolean }?,

  ## default value: true
  attribute insertHyperlinks { xsd:boolean }?,

  ## default value: true
  attribute deleteColumns { xsd:boolean }?,

  ## default value: true
  attribute deleteRows { xsd:boolean }?,

  ## default value: false
  attribute selectLockedCells { xsd:boolean }?,

  ## default value: true
  attribute sort { xsd:boolean }?,

  ## default value: true
  attribute autoFilter { xsd:boolean }?,

  ## default value: true
  attribute pivotTables { xsd:boolean }?,

  ## default value: false
  attribute selectUnlockedCells { xsd:boolean }?
sml_CT_ProtectedRanges =
  element protectedRange { sml_CT_ProtectedRange }+
sml_CT_ProtectedRange =
  attribute password { sml_ST_UnsignedShortHex }?,
  attribute sqref { sml_ST_Sqref },
  attribute name { s_ST_Xstring },
  attribute securityDescriptor { xsd:string }?,
  attribute algorithmName { s_ST_Xstring }?,
```

```
  attribute hashValue { xsd:base64Binary }?,
  attribute saltValue { xsd:base64Binary }?,
  attribute spinCount { xsd:unsignedInt }?,
  element securityDescriptor { xsd:string }*
sml_CT_Scenario =
  attribute name { s_ST_Xstring },

  ## default value: false
  attribute locked { xsd:boolean }?,

  ## default value: false
  attribute hidden { xsd:boolean }?,
  attribute count { xsd:unsignedInt }?,
  attribute user { s_ST_Xstring }?,
  attribute comment { s_ST_Xstring }?,
  element inputCells { sml_CT_InputCells }+
sml_CT_InputCells =
  attribute r { sml_ST_CellRef },

  ## default value: false
  attribute deleted { xsd:boolean }?,

  ## default value: false
  attribute undone { xsd:boolean }?,
  attribute val { s_ST_Xstring },
  attribute numFmtId { sml_ST_NumFmtId }?
sml_CT_CellWatches = element cellWatch { sml_CT_CellWatch }+
sml_CT_CellWatch = attribute r { sml_ST_CellRef }
sml_CT_Chartsheet =
  element sheetPr { sml_CT_ChartsheetPr }?,
  element sheetViews { sml_CT_ChartsheetViews },
  element sheetProtection { sml_CT_ChartsheetProtection }?,
  element customSheetViews { sml_CT_CustomChartsheetViews }?,
  element pageMargins { sml_CT_PageMargins }?,
  element pageSetup { sml_CT_CsPageSetup }?,
  element headerFooter { sml_CT_HeaderFooter }?,
  element drawing { sml_CT_Drawing },
  element legacyDrawing { sml_CT_LegacyDrawing }?,
  element legacyDrawingHF { sml_CT_LegacyDrawing }?,
  element drawingHF { sml_CT_DrawingHF }?,
  element picture { sml_CT_SheetBackgroundPicture }?,
  element webPublishItems { sml_CT_WebPublishItems }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_ChartsheetPr =

  ## default value: true
  attribute published { xsd:boolean }?,
  attribute codeName { xsd:string }?,
  element tabColor { sml_CT_Color }?
sml_CT_ChartsheetViews =
  element sheetView { sml_CT_ChartsheetView }+,
  element extLst { sml_CT_ExtensionList }?
sml_CT_ChartsheetView =
```



```
  ## default value: false
  attribute tabSelected { xsd:boolean }?,

  ## default value: 100
  attribute zoomScale { xsd:unsignedInt }?,
  attribute workbookViewId { xsd:unsignedInt },

  ## default value: false
  attribute zoomToFit { xsd:boolean }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_ChartsheetProtection =
  attribute password { sml_ST_UnsignedShortHex }?,
  attribute algorithmName { s_ST_Xstring }?,
  attribute hashValue { xsd:base64Binary }?,
  attribute saltValue { xsd:base64Binary }?,
  attribute spinCount { xsd:unsignedInt }?,

  ## default value: false
  attribute content { xsd:boolean }?,

  ## default value: false
  attribute objects { xsd:boolean }?
sml_CT_CsPageSetup =

  ## default value: 1
  attribute paperSize { xsd:unsignedInt }?,
  attribute paperHeight { s_ST_PositiveUniversalMeasure }?,
  attribute paperWidth { s_ST_PositiveUniversalMeasure }?,

  ## default value: 1
  attribute firstPageNumber { xsd:unsignedInt }?,

  ## default value: default
  attribute orientation { sml_ST_Orientation }?,

  ## default value: true
  attribute usePrinterDefaults { xsd:boolean }?,

  ## default value: false
  attribute blackAndWhite { xsd:boolean }?,

  ## default value: false
  attribute draft { xsd:boolean }?,

  ## default value: false
  attribute useFirstPageNumber { xsd:boolean }?,

  ## default value: 600
  attribute horizontalDpi { xsd:unsignedInt }?,

  ## default value: 600
  attribute verticalDpi { xsd:unsignedInt }?,
```

```

  ## default value: 1
  attribute copies { xsd:unsignedInt }?,
  r_id?
sml_CT_CustomChartsheetViews =
  element customSheetView { sml_CT_CustomChartsheetView }*
sml_CT_CustomChartsheetView =
  attribute guid { s_ST_Guid },

  ## default value: 100
  attribute scale { xsd:unsignedInt }?,

  ## default value: visible
  attribute state { sml_ST_SheetState }?,

  ## default value: false
  attribute zoomToFit { xsd:boolean }?,
  element pageMargins { sml_CT_PageMargins }?,
  element pageSetup { sml_CT_CsPageSetup }?,
  element headerFooter { sml_CT_HeaderFooter }?
sml_CT_CustomProperties = element customPr { sml_CT_CustomProperty }+
sml_CT_CustomProperty =
  attribute name { s_ST_Xstring },
  r_id
sml_CT_OleObjects = element oleObject { sml_CT_OleObject }+
sml_CT_OleObject =
  attribute progId { xsd:string }?,

  ## default value: DVASPECT_CONTENT
  attribute dvAspect { sml_ST_DvAspect }?,
  attribute link { s_ST_Xstring }?,
  attribute oleUpdate { sml_ST_OleUpdate }?,

  ## default value: false
  attribute autoLoad { xsd:boolean }?,
  attribute shapeId { xsd:unsignedInt },
  r_id?,
  element objectPr { sml_CT_ObjectPr }?
sml_CT_ObjectPr =

  ## default value: true
  attribute locked { xsd:boolean }?,

  ## default value: true
  attribute defaultSize { xsd:boolean }?,

  ## default value: true
  attribute print { xsd:boolean }?,

  ## default value: false
  attribute disabled { xsd:boolean }?,

  ## default value: false
```



```
  attribute uiObject { xsd:boolean }?,

  ## default value: true
  attribute autoFill { xsd:boolean }?,

  ## default value: true
  attribute autoLine { xsd:boolean }?,

  ## default value: true
  attribute autoPict { xsd:boolean }?,
  attribute macro { sml_ST_Formula }?,
  attribute altText { s_ST_Xstring }?,

  ## default value: false
  attribute dde { xsd:boolean }?,
  r_id?,
  element anchor { sml_CT_ObjectAnchor }
sml_ST_DvAspect = string "DVASPECT_CONTENT" | string "DVASPECT_ICON"
sml_ST_OleUpdate = string "OLEUPDATE_ALWAYS" | string "OLEUPDATE_ONCALL"
sml_CT_WebPublishItems =
  attribute count { xsd:unsignedInt }?,
  element webPublishItem { sml_CT_WebPublishItem }+
sml_CT_WebPublishItem =
  attribute id { xsd:unsignedInt },
  attribute divId { s_ST_Xstring },
  attribute sourceType { sml_ST_WebSourceType },
  attribute sourceRef { sml_ST_Ref }?,
  attribute sourceObject { s_ST_Xstring }?,
  attribute destinationFile { s_ST_Xstring },
  attribute title { s_ST_Xstring }?,

  ## default value: false
  attribute autoRepublish { xsd:boolean }?
sml_CT_Controls = element control { sml_CT_Control }+
sml_CT_Control =
  attribute shapeId { xsd:unsignedInt },
  r_id,
  attribute name { xsd:string }?,
  element controlPr { sml_CT_ControlPr }?
sml_CT_ControlPr =

  ## default value: true
  attribute locked { xsd:boolean }?,

  ## default value: true
  attribute defaultSize { xsd:boolean }?,

  ## default value: true
  attribute print { xsd:boolean }?,

  ## default value: false
  attribute disabled { xsd:boolean }?,

```

```
  ## default value: false
  attribute recalcAlways { xsd:boolean }?,

  ## default value: false
  attribute uiObject { xsd:boolean }?,

  ## default value: true
  attribute autoFill { xsd:boolean }?,

  ## default value: true
  attribute autoLine { xsd:boolean }?,

  ## default value: true
  attribute autoPict { xsd:boolean }?,
  attribute macro { sml_ST_Formula }?,
  attribute altText { s_ST_Xstring }?,
  attribute linkedCell { sml_ST_Formula }?,
  attribute listFillRange { sml_ST_Formula }?,

  ## default value: pict
  attribute cf { s_ST_Xstring }?,
  r_id?,
  element anchor { sml_CT_ObjectAnchor }
sml_ST_WebSourceType =
  string "sheet"
  | string "printArea"
  | string "autoFilter"
  | string "range"
  | string "chart"
  | string "pivotTable"
  | string "query"
  | string "label"
sml_CT_IgnoredErrors =
  element ignoredError { sml_CT_IgnoredError }+,
  element extLst { sml_CT_ExtensionList }?
sml_CT_IgnoredError =
  attribute sqref { sml_ST_Sqref },

  ## default value: false
  attribute evalError { xsd:boolean }?,

  ## default value: false
  attribute twoDigitTextYear { xsd:boolean }?,

  ## default value: false
  attribute numberStoredAsText { xsd:boolean }?,

  ## default value: false
  attribute formula { xsd:boolean }?,

  ## default value: false
  attribute formulaRange { xsd:boolean }?,

```



```
  ## default value: false
  attribute unlockedFormula { xsd:boolean }?,

  ## default value: false
  attribute emptyCellReference { xsd:boolean }?,

  ## default value: false
  attribute listDataValidation { xsd:boolean }?,

  ## default value: false
  attribute calculatedColumn { xsd:boolean }?
sml_ST_PaneState =
  string "split" | string "frozen" | string "frozenSplit"
sml_CT_TableParts =
  attribute count { xsd:unsignedInt }?,
  element tablePart { sml_CT_TablePart }*
sml_CT_TablePart = r_id
sml_metadata = element metadata { sml_CT_Metadata }
sml_CT_Metadata =
  element metadataTypes { sml_CT_MetadataTypes }?,
  element metadataStrings { sml_CT_MetadataStrings }?,
  element mdxMetadata { sml_CT_MdxMetadata }?,
  element futureMetadata { sml_CT_FutureMetadata }*,
  element cellMetadata { sml_CT_MetadataBlocks }?,
  element valueMetadata { sml_CT_MetadataBlocks }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_MetadataTypes =

  ## default value: 0
  attribute count { xsd:unsignedInt }?,
  element metadataType { sml_CT_MetadataType }+
sml_CT_MetadataType =
  attribute name { s_ST_Xstring },
  attribute minSupportedVersion { xsd:unsignedInt },

  ## default value: false
  attribute ghostRow { xsd:boolean }?,

  ## default value: false
  attribute ghostCol { xsd:boolean }?,

  ## default value: false
  attribute edit { xsd:boolean }?,

  ## default value: false
  attribute delete { xsd:boolean }?,

  ## default value: false
  attribute copy { xsd:boolean }?,

  ## default value: false
  attribute pasteAll { xsd:boolean }?,

```

```
  ## default value: false
  attribute pasteFormulas { xsd:boolean }?,

  ## default value: false
  attribute pasteValues { xsd:boolean }?,

  ## default value: false
  attribute pasteFormats { xsd:boolean }?,

  ## default value: false
  attribute pasteComments { xsd:boolean }?,

  ## default value: false
  attribute pasteDataValidation { xsd:boolean }?,

  ## default value: false
  attribute pasteBorders { xsd:boolean }?,

  ## default value: false
  attribute pasteColWidths { xsd:boolean }?,

  ## default value: false
  attribute pasteNumberFormats { xsd:boolean }?,

  ## default value: false
  attribute merge { xsd:boolean }?,

  ## default value: false
  attribute splitFirst { xsd:boolean }?,

  ## default value: false
  attribute splitAll { xsd:boolean }?,

  ## default value: false
  attribute rowColShift { xsd:boolean }?,

  ## default value: false
  attribute clearAll { xsd:boolean }?,

  ## default value: false
  attribute clearFormats { xsd:boolean }?,

  ## default value: false
  attribute clearContents { xsd:boolean }?,

  ## default value: false
  attribute clearComments { xsd:boolean }?,

  ## default value: false
  attribute assign { xsd:boolean }?,

  ## default value: false
  attribute coerce { xsd:boolean }?,
```



```
  ## default value: false
  attribute adjust { xsd:boolean }?,

  ## default value: false
  attribute cellMeta { xsd:boolean }?
sml_CT_MetadataBlocks =

  ## default value: 0
  attribute count { xsd:unsignedInt }?,
  element bk { sml_CT_MetadataBlock }+
sml_CT_MetadataBlock = element rc { sml_CT_MetadataRecord }+
sml_CT_MetadataRecord =
  attribute t { xsd:unsignedInt },
  attribute v { xsd:unsignedInt }
sml_CT_FutureMetadata =
  attribute name { s_ST_Xstring },

  ## default value: 0
  attribute count { xsd:unsignedInt }?,
  element bk { sml_CT_FutureMetadataBlock }*,
  element extLst { sml_CT_ExtensionList }?
sml_CT_FutureMetadataBlock = element extLst { sml_CT_ExtensionList }?
sml_CT_MdxMetadata =

  ## default value: 0
  attribute count { xsd:unsignedInt }?,
  element mdx { sml_CT_Mdx }+
sml_CT_Mdx =
  attribute n { xsd:unsignedInt },
  attribute f { sml_ST_MdxFunctionType },
  (element t { sml_CT_MdxTuple }
   | element ms { sml_CT_MdxSet }
   | element p { sml_CT_MdxMemeberProp }
   | element k { sml_CT_MdxKPI })
sml_ST_MdxFunctionType =
  string "m"
  | string "v"
  | string "s"
  | string "c"
  | string "r"
  | string "p"
  | string "k"
sml_CT_MdxTuple =

  ## default value: 0
  attribute c { xsd:unsignedInt }?,
  attribute ct { s_ST_Xstring }?,
  attribute si { xsd:unsignedInt }?,
  attribute fi { xsd:unsignedInt }?,
  attribute bc { sml_ST_UnsignedIntHex }?,
  attribute fc { sml_ST_UnsignedIntHex }?,

```

```
  ## default value: false
  attribute i { xsd:boolean }?,

  ## default value: false
  attribute u { xsd:boolean }?,

  ## default value: false
  attribute st { xsd:boolean }?,

  ## default value: false
  attribute b { xsd:boolean }?,
  element n { sml_CT_MetadataStringIndex }*
sml_CT_MdxSet =
  attribute ns { xsd:unsignedInt },

  ## default value: 0
  attribute c { xsd:unsignedInt }?,

  ## default value: u
  attribute o { sml_ST_MdxSetOrder }?,
  element n { sml_CT_MetadataStringIndex }*
sml_ST_MdxSetOrder =
  string "u"
  | string "a"
  | string "d"
  | string "aa"
  | string "ad"
  | string "na"
  | string "nd"
sml_CT_MdxMemeberProp =
  attribute n { xsd:unsignedInt },
  attribute np { xsd:unsignedInt }
sml_CT_MdxKPI =
  attribute n { xsd:unsignedInt },
  attribute np { xsd:unsignedInt },
  attribute p { sml_ST_MdxKPIProperty }
sml_ST_MdxKPIProperty =
  string "v"
  | string "g"
  | string "s"
  | string "t"
  | string "w"
  | string "m"
sml_CT_MetadataStringIndex =
  attribute x { xsd:unsignedInt },

  ## default value: false
  attribute s { xsd:boolean }?
sml_CT_MetadataStrings =

  ## default value: 0
  attribute count { xsd:unsignedInt }?,
  element s { sml_CT_XStringElement }+
```



```
sml_singleXmlCells = element singleXmlCells { sml_CT_SingleXmlCells }
sml_CT_SingleXmlCells = element singleXmlCell { sml_CT_SingleXmlCell }+
sml_CT_SingleXmlCell =
  attribute id { xsd:unsignedInt },
  attribute r { sml_ST_CellRef },
  attribute connectionId { xsd:unsignedInt },
  element xmlCellPr { sml_CT_XmlCellPr },
  element extLst { sml_CT_ExtensionList }?
sml_CT_XmlCellPr =
  attribute id { xsd:unsignedInt },
  attribute uniqueName { s_ST_Xstring }?,
  element xmlPr { sml_CT_XmlPr },
  element extLst { sml_CT_ExtensionList }?
sml_CT_XmlPr =
  attribute mapId { xsd:unsignedInt },
  attribute xpath { s_ST_Xstring },
  attribute xmlDataType { sml_ST_XmlDataType },
  element extLst { sml_CT_ExtensionList }?
sml_styleSheet = element styleSheet { sml_CT_Stylesheet }
sml_CT_Stylesheet =
  element numFmts { sml_CT_NumFmts }?,
  element fonts { sml_CT_Fonts }?,
  element fills { sml_CT_Fills }?,
  element borders { sml_CT_Borders }?,
  element cellStyleXfs { sml_CT_CellStyleXfs }?,
  element cellXfs { sml_CT_CellXfs }?,
  element cellStyles { sml_CT_CellStyles }?,
  element dxfs { sml_CT_Dxfs }?,
  element tableStyles { sml_CT_TableStyles }?,
  element colors { sml_CT_Colors }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_CellAlignment =
  attribute horizontal { sml_ST_HorizontalAlignment }?,
  attribute vertical { sml_ST_VerticalAlignment }?,
  attribute textRotation { xsd:unsignedInt }?,
  attribute wrapText { xsd:boolean }?,
  attribute indent { xsd:unsignedInt }?,
  attribute relativeIndent { xsd:int }?,
  attribute justifyLastLine { xsd:boolean }?,
  attribute shrinkToFit { xsd:boolean }?,
  attribute readingOrder { xsd:unsignedInt }?
sml_ST_BorderStyle =
  string "none"
  | string "thin"
  | string "medium"
  | string "dashed"
  | string "dotted"
  | string "thick"
  | string "double"
  | string "hair"
  | string "mediumDashed"
  | string "dashDot"
  | string "mediumDashDot"
```

```
  | string "dashDotDot"
  | string "mediumDashDotDot"
  | string "slantDashDot"
sml_CT_Borders =
  attribute count { xsd:unsignedInt }?,
  element border { sml_CT_Border }*
sml_CT_Border =
  attribute diagonalUp { xsd:boolean }?,
  attribute diagonalDown { xsd:boolean }?,

  ## default value: true
  attribute outline { xsd:boolean }?,
  element start { sml_CT_BorderPr }?,
  element end { sml_CT_BorderPr }?,
  element left { sml_CT_BorderPr }?,
  element right { sml_CT_BorderPr }?,
  element top { sml_CT_BorderPr }?,
  element bottom { sml_CT_BorderPr }?,
  element diagonal { sml_CT_BorderPr }?,
  element vertical { sml_CT_BorderPr }?,
  element horizontal { sml_CT_BorderPr }?
sml_CT_BorderPr =

  ## default value: none
  attribute style { sml_ST_BorderStyle }?,
  element color { sml_CT_Color }?
sml_CT_CellProtection =
  attribute locked { xsd:boolean }?,
  attribute hidden { xsd:boolean }?
sml_CT_Fonts =
  attribute count { xsd:unsignedInt }?,
  element font { sml_CT_Font }*
sml_CT_Fills =
  attribute count { xsd:unsignedInt }?,
  element fill { sml_CT_Fill }*
sml_CT_Fill =
  element patternFill { sml_CT_PatternFill }?
  | element gradientFill { sml_CT_GradientFill }?
sml_CT_PatternFill =
  attribute patternType { sml_ST_PatternType }?,
  element fgColor { sml_CT_Color }?,
  element bgColor { sml_CT_Color }?
sml_CT_Color =
  attribute auto { xsd:boolean }?,
  attribute indexed { xsd:unsignedInt }?,
  attribute rgb { sml_ST_UnsignedIntHex }?,
  attribute theme { xsd:unsignedInt }?,

  ## default value: 0.0
  attribute tint { xsd:double }?
sml_ST_PatternType =
  string "none"
  | string "solid"
```



```
  | string "mediumGray"
  | string "darkGray"
  | string "lightGray"
  | string "darkHorizontal"
  | string "darkVertical"
  | string "darkDown"
  | string "darkUp"
  | string "darkGrid"
  | string "darkTrellis"
  | string "lightHorizontal"
  | string "lightVertical"
  | string "lightDown"
  | string "lightUp"
  | string "lightGrid"
  | string "lightTrellis"
  | string "gray125"
  | string "gray0625"
sml_CT_GradientFill =

  ## default value: linear
  attribute type { sml_ST_GradientType }?,

  ## default value: 0
  attribute degree { xsd:double }?,

  ## default value: 0
  attribute left { xsd:double }?,

  ## default value: 0
  attribute right { xsd:double }?,

  ## default value: 0
  attribute top { xsd:double }?,

  ## default value: 0
  attribute bottom { xsd:double }?,
  element stop { sml_CT_GradientStop }*
sml_CT_GradientStop =
  attribute position { xsd:double },
  element color { sml_CT_Color }
sml_ST_GradientType = string "linear" | string "path"
sml_ST_HorizontalAlignment =
  string "general"
  | string "left"
  | string "center"
  | string "right"
  | string "fill"
  | string "justify"
  | string "centerContinuous"
  | string "distributed"
sml_ST_VerticalAlignment =
  string "top"
  | string "center"
```

```
  | string "bottom"
  | string "justify"
  | string "distributed"
sml_CT_NumFmts =
  attribute count { xsd:unsignedInt }?,
  element numFmt { sml_CT_NumFmt }*
sml_CT_NumFmt =
  attribute numFmtId { sml_ST_NumFmtId },
  attribute formatCode { s_ST_Xstring }
sml_CT_CellStyleXfs =
  attribute count { xsd:unsignedInt }?,
  element xf { sml_CT_Xf }+
sml_CT_CellXfs =
  attribute count { xsd:unsignedInt }?,
  element xf { sml_CT_Xf }+
sml_CT_Xf =
  attribute numFmtId { sml_ST_NumFmtId }?,
  attribute fontId { sml_ST_FontId }?,
  attribute fillId { sml_ST_FillId }?,
  attribute borderId { sml_ST_BorderId }?,
  attribute xfId { sml_ST_CellStyleXfId }?,

  ## default value: false
  attribute quotePrefix { xsd:boolean }?,

  ## default value: false
  attribute pivotButton { xsd:boolean }?,
  attribute applyNumberFormat { xsd:boolean }?,
  attribute applyFont { xsd:boolean }?,
  attribute applyFill { xsd:boolean }?,
  attribute applyBorder { xsd:boolean }?,
  attribute applyAlignment { xsd:boolean }?,
  attribute applyProtection { xsd:boolean }?,
  element alignment { sml_CT_CellAlignment }?,
  element protection { sml_CT_CellProtection }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_CellStyles =
  attribute count { xsd:unsignedInt }?,
  element cellStyle { sml_CT_CellStyle }+
sml_CT_CellStyle =
  attribute name { s_ST_Xstring }?,
  attribute xfId { sml_ST_CellStyleXfId },
  attribute builtinId { xsd:unsignedInt }?,
  attribute iLevel { xsd:unsignedInt }?,
  attribute hidden { xsd:boolean }?,
  attribute customBuiltin { xsd:boolean }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_Dxfs =
  attribute count { xsd:unsignedInt }?,
  element dxf { sml_CT_Dxf }*
sml_CT_Dxf =
  element font { sml_CT_Font }?,
  element numFmt { sml_CT_NumFmt }?,
```



```
  element fill { sml_CT_Fill }?,
  element alignment { sml_CT_CellAlignment }?,
  element border { sml_CT_Border }?,
  element protection { sml_CT_CellProtection }?,
  element extLst { sml_CT_ExtensionList }?
sml_ST_NumFmtId = xsd:unsignedInt
sml_ST_FontId = xsd:unsignedInt
sml_ST_FillId = xsd:unsignedInt
sml_ST_BorderId = xsd:unsignedInt
sml_ST_CellStyleXfId = xsd:unsignedInt
sml_ST_DxfId = xsd:unsignedInt
sml_CT_Colors =
  element indexedColors { sml_CT_IndexedColors }?,
  element mruColors { sml_CT_MRUColors }?
sml_CT_IndexedColors = element rgbColor { sml_CT_RgbColor }+
sml_CT_MRUColors = element color { sml_CT_Color }+
sml_CT_RgbColor = attribute rgb { sml_ST_UnsignedIntHex }?
sml_CT_TableStyles =
  attribute count { xsd:unsignedInt }?,
  attribute defaultTableStyle { xsd:string }?,
  attribute defaultPivotStyle { xsd:string }?,
  element tableStyle { sml_CT_TableStyle }*
sml_CT_TableStyle =
  attribute name { xsd:string },

  ## default value: true
  attribute pivot { xsd:boolean }?,

  ## default value: true
  attribute table { xsd:boolean }?,
  attribute count { xsd:unsignedInt }?,
  element tableStyleElement { sml_CT_TableStyleElement }*
sml_CT_TableStyleElement =
  attribute type { sml_ST_TableStyleType },

  ## default value: 1
  attribute size { xsd:unsignedInt }?,
  attribute dxfId { sml_ST_DxfId }?
sml_ST_TableStyleType =
  string "wholeTable"
  | string "headerRow"
  | string "totalRow"
  | string "firstColumn"
  | string "lastColumn"
  | string "firstRowStripe"
  | string "secondRowStripe"
  | string "firstColumnStripe"
  | string "secondColumnStripe"
  | string "firstHeaderCell"
  | string "lastHeaderCell"
  | string "firstTotalCell"
  | string "lastTotalCell"
  | string "firstSubtotalColumn"
```

```
  | string "secondSubtotalColumn"
  | string "thirdSubtotalColumn"
  | string "firstSubtotalRow"
  | string "secondSubtotalRow"
  | string "thirdSubtotalRow"
  | string "blankRow"
  | string "firstColumnSubheading"
  | string "secondColumnSubheading"
  | string "thirdColumnSubheading"
  | string "firstRowSubheading"
  | string "secondRowSubheading"
  | string "thirdRowSubheading"
  | string "pageFieldLabels"
  | string "pageFieldValues"
sml_CT_BooleanProperty =

  ## default value: true
  attribute val { xsd:boolean }?
sml_CT_FontSize = attribute val { xsd:double }
sml_CT_IntProperty = attribute val { xsd:int }
sml_CT_FontName = attribute val { s_ST_Xstring }
sml_CT_VerticalAlignFontProperty =
  attribute val { s_ST_VerticalAlignRun }
sml_CT_FontScheme = attribute val { sml_ST_FontScheme }
sml_ST_FontScheme = string "none" | string "major" | string "minor"
sml_CT_UnderlineProperty =

  ## default value: single
  attribute val { sml_ST_UnderlineValues }?
sml_ST_UnderlineValues =
  string "single"
  | string "double"
  | string "singleAccounting"
  | string "doubleAccounting"
  | string "none"
sml_CT_Font =
  (element name { sml_CT_FontName }?
   | element charset { sml_CT_IntProperty }?
   | element family { sml_CT_FontFamily}?
   | element b { sml_CT_BooleanProperty }?
   | element i { sml_CT_BooleanProperty }?
   | element strike { sml_CT_BooleanProperty }?
   | element outline { sml_CT_BooleanProperty }?
   | element shadow { sml_CT_BooleanProperty }?
   | element condense { sml_CT_BooleanProperty }?
   | element extend { sml_CT_BooleanProperty }?
   | element color { sml_CT_Color }?
   | element sz { sml_CT_FontSize }?
   | element u { sml_CT_UnderlineProperty }?
   | element vertAlign { sml_CT_VerticalAlignFontProperty }?
   | element scheme { sml_CT_FontScheme }?)+
sml_CT_FontFamily = attribute val { sml_ST_FontFamily }
sml_ST_FontFamily = xsd:integer { minInclusive = "0" maxInclusive = "14" }
```



```
sml_AG_AutoFormat =
  attribute autoFormatId { xsd:unsignedInt }?,
  attribute applyNumberFormats { xsd:boolean }?,
  attribute applyBorderFormats { xsd:boolean }?,
  attribute applyFontFormats { xsd:boolean }?,
  attribute applyPatternFormats { xsd:boolean }?,
  attribute applyAlignmentFormats { xsd:boolean }?,
  attribute applyWidthHeightFormats { xsd:boolean }?
sml_externalLink = element externalLink { sml_CT_ExternalLink }
sml_CT_ExternalLink =
  (element externalBook { sml_CT_ExternalBook }?
  | element ddeLink { sml_CT_DdeLink }?
  | element oleLink { sml_CT_OleLink }?),
  element extLst { sml_CT_ExtensionList }?
sml_CT_ExternalBook =
  r_id,
  element sheetNames { sml_CT_ExternalSheetNames }?,
  element definedNames { sml_CT_ExternalDefinedNames }?,
  element sheetDataSet { sml_CT_ExternalSheetDataSet }?
sml_CT_ExternalSheetNames =
  element sheetName { sml_CT_ExternalSheetName }+
sml_CT_ExternalSheetName = attribute val { s_ST_Xstring }?
sml_CT_ExternalDefinedNames =
  element definedName { sml_CT_ExternalDefinedName }*
sml_CT_ExternalDefinedName =
  attribute name { s_ST_Xstring },
  attribute refersTo { s_ST_Xstring }?,
  attribute sheetId { xsd:unsignedInt }?
sml_CT_ExternalSheetDataSet =
  element sheetData { sml_CT_ExternalSheetData }+
sml_CT_ExternalSheetData =
  attribute sheetId { xsd:unsignedInt },

  ## default value: false
  attribute refreshError { xsd:boolean }?,
  element row { sml_CT_ExternalRow }*
sml_CT_ExternalRow =
  attribute r { xsd:unsignedInt },
  element cell { sml_CT_ExternalCell }*
sml_CT_ExternalCell =
  attribute r { sml_ST_CellRef }?,

  ## default value: n
  attribute t { sml_ST_CellType }?,

  ## default value: 0
  attribute vm { xsd:unsignedInt }?,
  element v { s_ST_Xstring }?
sml_CT_DdeLink =
  attribute ddeService { s_ST_Xstring },
  attribute ddeTopic { s_ST_Xstring },
  element ddeItems { sml_CT_DdeItems }?
sml_CT_DdeItems = element ddeItem { sml_CT_DdeItem }*
```

```
sml_CT_DdeItem =

  ## default value: 0
  attribute name { s_ST_Xstring }?,

  ## default value: false
  attribute ole { xsd:boolean }?,

  ## default value: false
  attribute advise { xsd:boolean }?,

  ## default value: false
  attribute preferPic { xsd:boolean }?,
  element values { sml_CT_DdeValues }?
sml_CT_DdeValues =

  ## default value: 1
  attribute rows { xsd:unsignedInt }?,

  ## default value: 1
  attribute cols { xsd:unsignedInt }?,
  element value { sml_CT_DdeValue }+
sml_CT_DdeValue =

  ## default value: n
  attribute t { sml_ST_DdeValueType }?,
  element val { s_ST_Xstring }
sml_ST_DdeValueType =
  string "nil" | string "b" | string "n" | string "e" | string "str"
sml_CT_OleLink =
  r_id,
  attribute progId { s_ST_Xstring },
  element oleItems { sml_CT_OleItems }?
sml_CT_OleItems = element oleItem { sml_CT_OleItem }*
sml_CT_OleItem =
  attribute name { s_ST_Xstring },

  ## default value: false
  attribute icon { xsd:boolean }?,

  ## default value: false
  attribute advise { xsd:boolean }?,

  ## default value: false
  attribute preferPic { xsd:boolean }?
sml_table = element table { sml_CT_Table }
sml_CT_Table =
  attribute id { xsd:unsignedInt },
  attribute name { s_ST_Xstring }?,
  attribute displayName { s_ST_Xstring },
  attribute comment { s_ST_Xstring }?,
  attribute ref { sml_ST_Ref },

```



```
  ## default value: worksheet
  attribute tableType { sml_ST_TableType }?,

  ## default value: 1
  attribute headerRowCount { xsd:unsignedInt }?,

  ## default value: false
  attribute insertRow { xsd:boolean }?,

  ## default value: false
  attribute insertRowShift { xsd:boolean }?,

  ## default value: 0
  attribute totalsRowCount { xsd:unsignedInt }?,

  ## default value: true
  attribute totalsRowShown { xsd:boolean }?,

  ## default value: false
  attribute published { xsd:boolean }?,
  attribute headerRowDxfId { sml_ST_DxfId }?,
  attribute dataDxfId { sml_ST_DxfId }?,
  attribute totalsRowDxfId { sml_ST_DxfId }?,
  attribute headerRowBorderDxfId { sml_ST_DxfId }?,
  attribute tableBorderDxfId { sml_ST_DxfId }?,
  attribute totalsRowBorderDxfId { sml_ST_DxfId }?,
  attribute headerRowCellStyle { s_ST_Xstring }?,
  attribute dataCellStyle { s_ST_Xstring }?,
  attribute totalsRowCellStyle { s_ST_Xstring }?,
  attribute connectionId { xsd:unsignedInt }?,
  element autoFilter { sml_CT_AutoFilter }?,
  element sortState { sml_CT_SortState }?,
  element tableColumns { sml_CT_TableColumns },
  element tableStyleInfo { sml_CT_TableStyleInfo }?,
  element extLst { sml_CT_ExtensionList }?
sml_ST_TableType =
  string "worksheet" | string "xml" | string "queryTable"
sml_CT_TableStyleInfo =
  attribute name { s_ST_Xstring }?,
  attribute showFirstColumn { xsd:boolean }?,
  attribute showLastColumn { xsd:boolean }?,
  attribute showRowStripes { xsd:boolean }?,
  attribute showColumnStripes { xsd:boolean }?
sml_CT_TableColumns =
  attribute count { xsd:unsignedInt }?,
  element tableColumn { sml_CT_TableColumn }+
sml_CT_TableColumn =
  attribute id { xsd:unsignedInt },
  attribute uniqueName { s_ST_Xstring }?,
  attribute name { s_ST_Xstring },

  ## default value: none
  attribute totalsRowFunction { sml_ST_TotalsRowFunction }?,
```

```
  attribute totalsRowLabel { s_ST_Xstring }?,
  attribute queryTableFieldId { xsd:unsignedInt }?,
  attribute headerRowDxfId { sml_ST_DxfId }?,
  attribute dataDxfId { sml_ST_DxfId }?,
  attribute totalsRowDxfId { sml_ST_DxfId }?,
  attribute headerRowCellStyle { s_ST_Xstring }?,
  attribute dataCellStyle { s_ST_Xstring }?,
  attribute totalsRowCellStyle { s_ST_Xstring }?,
  element calculatedColumnFormula { sml_CT_TableFormula }?,
  element totalsRowFormula { sml_CT_TableFormula }?,
  element xmlColumnPr { sml_CT_XmlColumnPr }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_TableFormula =
  sml_ST_Formula,

  ## default value: false
  attribute array { xsd:boolean }?
sml_ST_TotalsRowFunction =
  string "none"
  | string "sum"
  | string "min"
  | string "max"
  | string "average"
  | string "count"
  | string "countNums"
  | string "stdDev"
  | string "var"
  | string "custom"
sml_CT_XmlColumnPr =
  attribute mapId { xsd:unsignedInt },
  attribute xpath { s_ST_Xstring },

  ## default value: false
  attribute denormalized { xsd:boolean }?,
  attribute xmlDataType { sml_ST_XmlDataType },
  element extLst { sml_CT_ExtensionList }?
sml_ST_XmlDataType = xsd:string
sml_volTypes = element volTypes { sml_CT_VolTypes }
sml_CT_VolTypes =
  element volType { sml_CT_VolType }+,
  element extLst { sml_CT_ExtensionList }?
sml_CT_VolType =
  attribute type { sml_ST_VolDepType },
  element main { sml_CT_VolMain }+
sml_CT_VolMain =
  attribute first { s_ST_Xstring },
  element tp { sml_CT_VolTopic }+
sml_CT_VolTopic =

  ## default value: n
  attribute t { sml_ST_VolValueType }?,
  element v { s_ST_Xstring },
  element stp { s_ST_Xstring }*,
```



```
  element tr { sml_CT_VolTopicRef }+
sml_CT_VolTopicRef =
  attribute r { sml_ST_CellRef },
  attribute s { xsd:unsignedInt }
sml_ST_VolDepType = string "realTimeData" | string "olapFunctions"
sml_ST_VolValueType = string "b" | string "n" | string "e" | string "s"
sml_workbook = element workbook { sml_CT_Workbook }
sml_CT_Workbook =
  attribute conformance { s_ST_ConformanceClass }?,
  element fileVersion { sml_CT_FileVersion }?,
  element fileSharing { sml_CT_FileSharing }?,
  element workbookPr { sml_CT_WorkbookPr }?,
  element workbookProtection { sml_CT_WorkbookProtection }?,
  element bookViews { sml_CT_BookViews }?,
  element sheets { sml_CT_Sheets },
  element functionGroups { sml_CT_FunctionGroups }?,
  element externalReferences { sml_CT_ExternalReferences }?,
  element definedNames { sml_CT_DefinedNames }?,
  element calcPr { sml_CT_CalcPr }?,
  element oleSize { sml_CT_OleSize }?,
  element customWorkbookViews { sml_CT_CustomWorkbookViews }?,
  element pivotCaches { sml_CT_PivotCaches }?,
  element smartTagPr { sml_CT_SmartTagPr }?,
  element smartTagTypes { sml_CT_SmartTagTypes }?,
  element webPublishing { sml_CT_WebPublishing }?,
  element fileRecoveryPr { sml_CT_FileRecoveryPr }*,
  element webPublishObjects { sml_CT_WebPublishObjects }?,
  element extLst { sml_CT_ExtensionList }?
sml_CT_FileVersion =
  attribute appName { xsd:string }?,
  attribute lastEdited { xsd:string }?,
  attribute lowestEdited { xsd:string }?,
  attribute rupBuild { xsd:string }?,
  attribute codeName { s_ST_Guid }?
sml_CT_BookViews = element workbookView { sml_CT_BookView }+
sml_CT_BookView =

  ## default value: visible
  attribute visibility { sml_ST_Visibility }?,

  ## default value: false
  attribute minimized { xsd:boolean }?,

  ## default value: true
  attribute showHorizontalScroll { xsd:boolean }?,

  ## default value: true
  attribute showVerticalScroll { xsd:boolean }?,

  ## default value: true
  attribute showSheetTabs { xsd:boolean }?,
  attribute xWindow { xsd:int }?,
  attribute yWindow { xsd:int }?,
```

```
  attribute windowWidth { xsd:unsignedInt }?,
  attribute windowHeight { xsd:unsignedInt }?,

  ## default value: 600
  attribute tabRatio { xsd:unsignedInt }?,

  ## default value: 0
  attribute firstSheet { xsd:unsignedInt }?,

  ## default value: 0
  attribute activeTab { xsd:unsignedInt }?,

  ## default value: true
  attribute autoFilterDateGrouping { xsd:boolean }?,
  element extLst { sml_CT_ExtensionList }?
sml_ST_Visibility =
  string "visible" | string "hidden" | string "veryHidden"
sml_CT_CustomWorkbookViews =
  element customWorkbookView { sml_CT_CustomWorkbookView }+
sml_CT_CustomWorkbookView =
  attribute name { s_ST_Xstring },
  attribute guid { s_ST_Guid },

  ## default value: false
  attribute autoUpdate { xsd:boolean }?,
  attribute mergeInterval { xsd:unsignedInt }?,

  ## default value: false
  attribute changesSavedWin { xsd:boolean }?,

  ## default value: false
  attribute onlySync { xsd:boolean }?,

  ## default value: false
  attribute personalView { xsd:boolean }?,

  ## default value: true
  attribute includePrintSettings { xsd:boolean }?,

  ## default value: true
  attribute includeHiddenRowCol { xsd:boolean }?,

  ## default value: false
  attribute maximized { xsd:boolean }?,

  ## default value: false
  attribute minimized { xsd:boolean }?,

  ## default value: true
  attribute showHorizontalScroll { xsd:boolean }?,

  ## default value: true
  attribute showVerticalScroll { xsd:boolean }?,
```



```
  ## default value: true
  attribute showSheetTabs { xsd:boolean }?,

  ## default value: 0
  attribute xWindow { xsd:int }?,

  ## default value: 0
  attribute yWindow { xsd:int }?,
  attribute windowWidth { xsd:unsignedInt },
  attribute windowHeight { xsd:unsignedInt },

  ## default value: 600
  attribute tabRatio { xsd:unsignedInt }?,
  attribute activeSheetId { xsd:unsignedInt },

  ## default value: true
  attribute showFormulaBar { xsd:boolean }?,

  ## default value: true
  attribute showStatusbar { xsd:boolean }?,

  ## default value: commIndicator
  attribute showComments { sml_ST_Comments }?,

  ## default value: all
  attribute showObjects { sml_ST_Objects }?,
  element extLst { sml_CT_ExtensionList }?
sml_ST_Comments =
  string "commNone"
  | string "commIndicator"
  | string "commIndAndComment"
sml_ST_Objects = string "all" | string "placeholders" | string "none"
sml_CT_Sheets = element sheet { sml_CT_Sheet }+
sml_CT_Sheet =
  attribute name { s_ST_Xstring },
  attribute sheetId { xsd:unsignedInt },

  ## default value: visible
  attribute state { sml_ST_SheetState }?,
  r_id
sml_ST_SheetState =
  string "visible" | string "hidden" | string "veryHidden"
sml_CT_WorkbookPr =

  ## default value: false
  attribute date1904 { xsd:boolean }?,

  ## default value: all
  attribute showObjects { sml_ST_Objects }?,

  ## default value: true
  attribute showBorderUnselectedTables { xsd:boolean }?,
```

```

  ## default value: false
  attribute filterPrivacy { xsd:boolean }?,

  ## default value: false
  attribute promptedSolutions { xsd:boolean }?,

  ## default value: true
  attribute showInkAnnotation { xsd:boolean }?,

  ## default value: false
  attribute backupFile { xsd:boolean }?,

  ## default value: true
  attribute saveExternalLinkValues { xsd:boolean }?,

  ## default value: userSet
  attribute updateLinks { sml_ST_UpdateLinks }?,
  attribute codeName { xsd:string }?,

  ## default value: false
  attribute hidePivotFieldList { xsd:boolean }?,

  ## default value: false
  attribute showPivotChartFilter { xsd:boolean }?,

  ## default value: false
  attribute allowRefreshQuery { xsd:boolean }?,

  ## default value: false
  attribute publishItems { xsd:boolean }?,

  ## default value: false
  attribute checkCompatibility { xsd:boolean }?,

  ## default value: true
  attribute autoCompressPictures { xsd:boolean }?,

  ## default value: false
  attribute refreshAllConnections { xsd:boolean }?,
  attribute defaultThemeVersion { xsd:unsignedInt }?
sml_ST_UpdateLinks = string "userSet" | string "never" | string "always"
sml_CT_SmartTagPr =

  ## default value: false
  attribute embed { xsd:boolean }?,

  ## default value: all
  attribute show { sml_ST_SmartTagShow }?
sml_ST_SmartTagShow =
  string "all" | string "none" | string "noIndicator"
sml_CT_SmartTagTypes = element smartTagType { sml_CT_SmartTagType }*
sml_CT_SmartTagType =
```



```
  attribute namespaceUri { s_ST_Xstring }?,
  attribute name { s_ST_Xstring }?,
  attribute url { s_ST_Xstring }?
sml_CT_FileRecoveryPr =

  ## default value: true
  attribute autoRecover { xsd:boolean }?,

  ## default value: false
  attribute crashSave { xsd:boolean }?,

  ## default value: false
  attribute dataExtractLoad { xsd:boolean }?,

  ## default value: false
  attribute repairLoad { xsd:boolean }?
sml_CT_CalcPr =
  attribute calcId { xsd:unsignedInt }?,

  ## default value: auto
  attribute calcMode { sml_ST_CalcMode }?,

  ## default value: false
  attribute fullCalcOnLoad { xsd:boolean }?,

  ## default value: A1
  attribute refMode { sml_ST_RefMode }?,

  ## default value: false
  attribute iterate { xsd:boolean }?,

  ## default value: 100
  attribute iterateCount { xsd:unsignedInt }?,

  ## default value: 0.001
  attribute iterateDelta { xsd:double }?,

  ## default value: true
  attribute fullPrecision { xsd:boolean }?,

  ## default value: true
  attribute calcCompleted { xsd:boolean }?,

  ## default value: true
  attribute calcOnSave { xsd:boolean }?,

  ## default value: true
  attribute concurrentCalc { xsd:boolean }?,
  attribute concurrentManualCount { xsd:unsignedInt }?,
  attribute forceFullCalc { xsd:boolean }?
sml_ST_CalcMode = string "manual" | string "auto" | string "autoNoTable"
sml_ST_RefMode = string "A1" | string "R1C1"
sml_CT_DefinedNames = element definedName { sml_CT_DefinedName }*
```

```
sml_CT_DefinedName =
  sml_ST_Formula,
  attribute name { s_ST_Xstring },
  attribute comment { s_ST_Xstring }?,
  attribute customMenu { s_ST_Xstring }?,
  attribute description { s_ST_Xstring }?,
  attribute help { s_ST_Xstring }?,
  attribute statusBar { s_ST_Xstring }?,
  attribute localSheetId { xsd:unsignedInt }?,

  ## default value: false
  attribute hidden { xsd:boolean }?,

  ## default value: false
  attribute function { xsd:boolean }?,

  ## default value: false
  attribute vbProcedure { xsd:boolean }?,

  ## default value: false
  attribute xlm { xsd:boolean }?,
  attribute functionGroupId { xsd:unsignedInt }?,
  attribute shortcutKey { s_ST_Xstring }?,

  ## default value: false
  attribute publishToServer { xsd:boolean }?,

  ## default value: false
  attribute workbookParameter { xsd:boolean }?
sml_CT_ExternalReferences =
  element externalReference { sml_CT_ExternalReference }+
sml_CT_ExternalReference = r_id
sml_CT_SheetBackgroundPicture = r_id
sml_CT_PivotCaches = element pivotCache { sml_CT_PivotCache }+
sml_CT_PivotCache =
  attribute cacheId { xsd:unsignedInt },
  r_id
sml_CT_FileSharing =

  ## default value: false
  attribute readOnlyRecommended { xsd:boolean }?,
  attribute userName { s_ST_Xstring }?,
  attribute reservationPassword { sml_ST_UnsignedShortHex }?,
  attribute algorithmName { s_ST_Xstring }?,
  attribute hashValue { xsd:base64Binary }?,
  attribute saltValue { xsd:base64Binary }?,
  attribute spinCount { xsd:unsignedInt }?
sml_CT_OleSize = attribute ref { sml_ST_Ref }
sml_CT_WorkbookProtection =
  attribute workbookPassword { sml_ST_UnsignedShortHex }?,
  attribute workbookPasswordCharacterSet { xsd:string }?,
  attribute revisionsPassword { sml_ST_UnsignedShortHex }?,
  attribute revisionsPasswordCharacterSet { xsd:string }?,
```



```
  ## default value: false
  attribute lockStructure { xsd:boolean }?,

  ## default value: false
  attribute lockWindows { xsd:boolean }?,

  ## default value: false
  attribute lockRevision { xsd:boolean }?,
  attribute revisionsAlgorithmName { s_ST_Xstring }?,
  attribute revisionsHashValue { xsd:base64Binary }?,
  attribute revisionsSaltValue { xsd:base64Binary }?,
  attribute revisionsSpinCount { xsd:unsignedInt }?,
  attribute workbookAlgorithmName { s_ST_Xstring }?,
  attribute workbookHashValue { xsd:base64Binary }?,
  attribute workbookSaltValue { xsd:base64Binary }?,
  attribute workbookSpinCount { xsd:unsignedInt }?
sml_CT_WebPublishing =

  ## default value: true
  attribute css { xsd:boolean }?,

  ## default value: true
  attribute thicket { xsd:boolean }?,

  ## default value: true
  attribute longFileNames { xsd:boolean }?,

  ## default value: false
  attribute vml { xsd:boolean }?,

  ## default value: false
  attribute allowPng { xsd:boolean }?,

  ## default value: 800x600
  attribute targetScreenSize { sml_ST_TargetScreenSize }?,

  ## default value: 96
  attribute dpi { xsd:unsignedInt }?,
  attribute codePage { xsd:unsignedInt }?,
  attribute characterSet { xsd:string }?
sml_ST_TargetScreenSize =
  string "544x376"
  | string "640x480"
  | string "720x512"
  | string "800x600"
  | string "1024x768"
  | string "1152x882"
  | string "1152x900"
  | string "1280x1024"
  | string "1600x1200"
  | string "1800x1440"
  | string "1920x1200"
```

```
sml_CT_FunctionGroups =

  ## default value: 16
  attribute builtInGroupCount { xsd:unsignedInt }?,
  element functionGroup { sml_CT_FunctionGroup }*
sml_CT_FunctionGroup = attribute name { s_ST_Xstring }?
sml_CT_WebPublishObjects =
  attribute count { xsd:unsignedInt }?,
  element webPublishObject { sml_CT_WebPublishObject }+
sml_CT_WebPublishObject =
  attribute id { xsd:unsignedInt },
  attribute divId { s_ST_Xstring },
  attribute sourceObject { s_ST_Xstring }?,
  attribute destinationFile { s_ST_Xstring },
  attribute title { s_ST_Xstring }?,

  ## default value: false
  attribute autoRepublish { xsd:boolean }?
```

B.2.1 Part Schemas

B.2.1.1 Calculation Chain Part

This schema is available in the file SpreadsheetML_Calculation_Chain.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = sml_calcChain
```

B.2.1.2 Chartsheet Part

This schema is available in the file SpreadsheetML_Chartsheet.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
```



```
start = sml_chartsheet
```

B.2.1.3 Comments Part

This schema is available in the file SpreadsheetML_Comments.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = sml_comments
```

B.2.1.4 Connections Part

This schema is available in the file SpreadsheetML_Connections.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = sml_connections
```

B.2.1.5 Custom XML Mappings Part

This schema is available in the file SpreadsheetML_Custom_XML_Mappings.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = sml_MapInfo
```

B.2.1.6 Dialogsheet Part

This schema is available in the file SpreadsheetML_Dialogsheet.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = sml_dialogsheet
```

B.2.1.7 Drawing Part

This schema is available in the file SpreadsheetML_Drawing.rnc.

```
include "dml-spreadsheetDrawing.rnc"
include "shared-relationshipReference.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = xdr_wsDr
```

B.2.1.8 External Workbook References Part

This schema is available in the file SpreadsheetML_External_Workbook_References.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = sml_externalLink
```

B.2.1.9 Metadata Part

This schema is available in the file SpreadsheetML_Metadata.rnc.


```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = sml_metadata
```

B.2.1.10 Pivot Table Part

This schema is available in the file SpreadsheetML_Pivot_Table.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = sml_pivotTableDefinition
```

B.2.1.11 Pivot Table Cache Definition Part

This schema is available in the file SpreadsheetML_Pivot_Table_Cache_Definition.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = sml_pivotCacheDefinition
```

B.2.1.12 Pivot Table Cache Records Part

This schema is available in the file SpreadsheetML_Pivot_Table_Cache_Records.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
```

```
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = sml_pivotCacheRecords
```

B.2.1.13 Query Table Part

This schema is available in the file SpreadsheetML_Query_Table.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = sml_queryTable
```

B.2.1.14 Shared String Table Part

This schema is available in the file SpreadsheetML_Shared_String_Table.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
include "dml-compatibility.rnc"
start = sml_sst
```

B.2.1.15 Shared Workbook Revision Headers Part

This schema is available in the file SpreadsheetML_Shared_Workbook_Revision_Headers.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
```



```
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = sml_headers
```

B.2.1.16 Shared Workbook Revision Log Part

This schema is available in the file SpreadsheetML_Shared_Workbook_Revision_Log.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = sml_revisions
```

B.2.1.17 Shared Workbook User Data Part

This schema is available in the file SpreadsheetML_Shared_Workbook_User_Data.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = sml_users
```

B.2.1.18 Single Cell Table Definitions Part

This schema is available in the file SpreadsheetML_Single_Cell_Table_Definitions.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
```

```
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = sml_singleXmlCells
```

B.2.1.19 Styles Part

This schema is available in the file SpreadsheetML_Styles.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = sml_styleSheet
```

B.2.1.20 Table Definitions Part

This schema is available in the file SpreadsheetML_Table_Definitions.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = sml_table
```

B.2.1.21 Volatile Dependencies Part

This schema is available in the file SpreadsheetML_Volatile_Dependencies.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
```



```
start = sml_volTypes
```

B.2.1.22 Workbook Part

This schema is available in the file SpreadsheetML_Workbook.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = sml_workbook
```

B.2.1.23 Worksheet Part

This schema is available in the file SpreadsheetML_Worksheet.rnc.

```
include "sml.rnc"
include "shared-relationshipReference.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-spreadsheetDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "dml-lockedCanvas.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = sml_worksheet
```

B.3 PresentationML

This schema is available in the file pml.rnc.

```
default namespace =
  "http://schemas.openxmlformats.org/presentationml/2006/main"
namespace a = "http://schemas.openxmlformats.org/drawingml/2006/main"
namespace o = "urn:schemas-microsoft-com:office:office"
namespace p =
  "http://schemas.openxmlformats.org/presentationml/2006/main"
namespace r =
  "http://schemas.openxmlformats.org/officeDocument/2006/relationships"
namespace s =
  "http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
namespace v = "urn:schemas-microsoft-com:vml"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"

```

```
p_ST_TransitionSideDirectionType = "l" | "u" | "r" | "d"
p_ST_TransitionCornerDirectionType = "lu" | "ru" | "ld" | "rd"
p_ST_TransitionInOutDirectionType = "out" | "in"
p_CT_SideDirectionTransition =

  ## default value: l
  attribute dir { p_ST_TransitionSideDirectionType }?
p_CT_CornerDirectionTransition =

  ## default value: lu
  attribute dir { p_ST_TransitionCornerDirectionType }?
p_ST_TransitionEightDirectionType =
  p_ST_TransitionSideDirectionType | p_ST_TransitionCornerDirectionType
p_CT_EightDirectionTransition =

  ## default value: l
  attribute dir { p_ST_TransitionEightDirectionType }?
p_CT_OrientationTransition =

  ## default value: horz
  attribute dir { p_ST_Direction }?
p_CT_InOutTransition =

  ## default value: out
  attribute dir { p_ST_TransitionInOutDirectionType }?
p_CT_OptionalBlackTransition =

  ## default value: false
  attribute thruBlk { xsd:boolean }?
p_CT_SplitTransition =

  ## default value: horz
  attribute orient { p_ST_Direction }?,

  ## default value: out
  attribute dir { p_ST_TransitionInOutDirectionType }?
p_CT_WheelTransition =

  ## default value: 4
  attribute spokes { xsd:unsignedInt }?
p_CT_TransitionStartSoundAction =

  ## default value: false
  attribute loop { xsd:boolean }?,
  element snd { a_CT_EmbeddedWAVAudioFile }
p_CT_TransitionSoundAction =
  element stSnd { p_CT_TransitionStartSoundAction }
  | element endSnd { p_CT_Empty }
p_ST_TransitionSpeed = "slow" | "med" | "fast"
p_CT_SlideTransition =

  ## default value: fast
  attribute spd { p_ST_TransitionSpeed }?,
```



```
  ## default value: true
  attribute advClick { xsd:boolean }?,
  attribute advTm { xsd:unsignedInt }?,
  (element blinds { p_CT_OrientationTransition }
   | element checker { p_CT_OrientationTransition }
   | element circle { p_CT_Empty }
   | element dissolve { p_CT_Empty }
   | element comb { p_CT_OrientationTransition }
   | element cover { p_CT_EightDirectionTransition }
   | element cut { p_CT_OptionalBlackTransition }
   | element diamond { p_CT_Empty }
   | element fade { p_CT_OptionalBlackTransition }
   | element newsflash { p_CT_Empty }
   | element plus { p_CT_Empty }
   | element pull { p_CT_EightDirectionTransition }
   | element push { p_CT_SideDirectionTransition }
   | element random { p_CT_Empty }
   | element randomBar { p_CT_OrientationTransition }
   | element split { p_CT_SplitTransition }
   | element strips { p_CT_CornerDirectionTransition }
   | element wedge { p_CT_Empty }
   | element wheel { p_CT_WheelTransition }
   | element wipe { p_CT_SideDirectionTransition }
   | element zoom { p_CT_InOutTransition })?,
  element sndAc { p_CT_TransitionSoundAction }?,
  element extLst { p_CT_ExtensionListModify }?
p_ST_TLTimeIndefinite = "indefinite"
p_ST_TLTime = xsd:unsignedInt | p_ST_TLTimeIndefinite
p_ST_TLTimeNodeID = xsd:unsignedInt
p_CT_TLIterateIntervalTime = attribute val { p_ST_TLTime }
p_CT_TLIterateIntervalPercentage =
  attribute val { a_ST_PositivePercentage }
p_ST_IterateType = "el" | "wd" | "lt"
p_CT_TLIterateData =

  ## default value: el
  attribute type { p_ST_IterateType }?,

  ## default value: false
  attribute backwards { xsd:boolean }?,
  (element tmAbs { p_CT_TLIterateIntervalTime }
   | element tmPct { p_CT_TLIterateIntervalPercentage })
p_CT_TLSubShapeId = attribute spid { a_ST_ShapeID }
p_CT_TLTextTargetElement =
  (element charRg { p_CT_IndexRange }
   | element pRg { p_CT_IndexRange })?
p_ST_TLChartSubelementType =
  "gridLegend" | "series" | "category" | "ptInSeries" | "ptInCategory"
p_CT_TLOleChartTargetElement =
  attribute type { p_ST_TLChartSubelementType },

  ## default value: 0
```

```
  attribute lvl { xsd:unsignedInt }?
p_CT_TLShapeTargetElement =
  attribute spid { a_ST_DrawingElementId },
  (element bg { p_CT_Empty }
   | element subSp { p_CT_TLSubShapeId }
   | element oleChartEl { p_CT_TLOleChartTargetElement }
   | element txEl { p_CT_TLTextTargetElement }
   | element graphicEl { a_CT_AnimationElementChoice })?
p_CT_TLTimeTargetElement =
  element sldTgt { p_CT_Empty }
  | element sndTgt { a_CT_EmbeddedWAVAudioFile }
  | element spTgt { p_CT_TLShapeTargetElement }
  | element inkTgt { p_CT_TLSubShapeId }
p_CT_TLTriggerTimeNodeID = attribute val { p_ST_TLTimeNodeID }
p_ST_TLTriggerRuntimeNode = "first" | "last" | "all"
p_CT_TLTriggerRuntimeNode = attribute val { p_ST_TLTriggerRuntimeNode }
p_ST_TLTriggerEvent =
  "onBegin"
  | "onEnd"
  | "begin"
  | "end"
  | "onClick"
  | "onDblClick"
  | "onMouseOver"
  | "onMouseOut"
  | "onNext"
  | "onPrev"
  | "onStopAudio"
p_CT_TLTimeCondition =
  attribute evt { p_ST_TLTriggerEvent }?,
  attribute delay { p_ST_TLTime }?,
  (element tgtEl { p_CT_TLTimeTargetElement }
   | element tn { p_CT_TLTriggerTimeNodeID }
   | element rtn { p_CT_TLTriggerRuntimeNode })?
p_CT_TLTimeConditionList = element cond { p_CT_TLTimeCondition }+
p_CT_TimeNodeList =
  (element par { p_CT_TLTimeNodeParallel }
   | element seq { p_CT_TLTimeNodeSequence }
   | element excl { p_CT_TLTimeNodeExclusive }
   | element anim { p_CT_TLAnimateBehavior }
   | element animClr { p_CT_TLAnimateColorBehavior }
   | element animEffect { p_CT_TLAnimateEffectBehavior }
   | element animMotion { p_CT_TLAnimateMotionBehavior }
   | element animRot { p_CT_TLAnimateRotationBehavior }
   | element animScale { p_CT_TLAnimateScaleBehavior }
   | element cmd { p_CT_TLCommandBehavior }
   | element set { p_CT_TLSetBehavior }
   | element audio { p_CT_TLMediaNodeAudio }
   | element video { p_CT_TLMediaNodeVideo })+
p_ST_TLTimeNodePresetClassType =
  "entr" | "exit" | "emph" | "path" | "verb" | "mediacall"
p_ST_TLTimeNodeRestartType = "always" | "whenNotActive" | "never"
p_ST_TLTimeNodeFillType = "remove" | "freeze" | "hold" | "transition"
```



```
p_ST_TLTimeNodeSyncType = "canSlip" | "locked"
p_ST_TLTimeNodeMasterRelation = "sameClick" | "lastClick" | "nextClick"
p_ST_TLTimeNodeType =
  "clickEffect"
  | "withEffect"
  | "afterEffect"
  | "mainSeq"
  | "interactiveSeq"
  | "clickPar"
  | "withGroup"
  | "afterGroup"
  | "tmRoot"
p_CT_TLCommonTimeNodeData =
  attribute id { p_ST_TLTimeNodeID }?,
  attribute presetID { xsd:int }?,
  attribute presetClass { p_ST_TLTimeNodePresetClassType }?,
  attribute presetSubtype { xsd:int }?,
  attribute dur { p_ST_TLTime }?,

  ## default value: 1000
  attribute repeatCount { p_ST_TLTime }?,
  attribute repeatDur { p_ST_TLTime }?,

  ## default value: 100%
  attribute spd { a_ST_Percentage }?,

  ## default value: 0%
  attribute accel { a_ST_PositiveFixedPercentage }?,

  ## default value: 0%
  attribute decel { a_ST_PositiveFixedPercentage }?,

  ## default value: false
  attribute autoRev { xsd:boolean }?,
  attribute restart { p_ST_TLTimeNodeRestartType }?,
  attribute fill { p_ST_TLTimeNodeFillType }?,
  attribute syncBehavior { p_ST_TLTimeNodeSyncType }?,
  attribute tmFilter { xsd:string }?,
  attribute evtFilter { xsd:string }?,
  attribute display { xsd:boolean }?,
  attribute masterRel { p_ST_TLTimeNodeMasterRelation }?,
  attribute bldLvl { xsd:int }?,
  attribute grpId { xsd:unsignedInt }?,
  attribute afterEffect { xsd:boolean }?,
  attribute nodeType { p_ST_TLTimeNodeType }?,
  attribute nodePh { xsd:boolean }?,
  element stCondLst { p_CT_TLTimeConditionList }?,
  element endCondLst { p_CT_TLTimeConditionList }?,
  element endSync { p_CT_TLTimeCondition }?,
  element iterate { p_CT_TLIterateData }?,
  element childTnLst { p_CT_TimeNodeList }?,
  element subTnLst { p_CT_TimeNodeList }?
p_CT_TLTimeNodeParallel = element cTn { p_CT_TLCommonTimeNodeData }
```

```
p_ST_TLNextActionType = "none" | "seek"
p_ST_TLPreviousActionType = "none" | "skipTimed"
p_CT_TLTimeNodeSequence =
  attribute concurrent { xsd:boolean }?,
  attribute prevAc { p_ST_TLPreviousActionType }?,
  attribute nextAc { p_ST_TLNextActionType }?,
  element cTn { p_CT_TLCommonTimeNodeData },
  element prevCondLst { p_CT_TLTimeConditionList }?,
  element nextCondLst { p_CT_TLTimeConditionList }?
p_CT_TLTimeNodeExclusive = element cTn { p_CT_TLCommonTimeNodeData }
p_CT_TLBehaviorAttributeNameList = element attrName { xsd:string }+
p_ST_TLBehaviorAdditiveType = "base" | "sum" | "repl" | "mult" | "none"
p_ST_TLBehaviorAccumulateType = "none" | "always"
p_ST_TLBehaviorTransformType = "pt" | "img"
p_ST_TLBehaviorOverrideType = "normal" | "childStyle"
p_CT_TLCommonBehaviorData =
  attribute additive { p_ST_TLBehaviorAdditiveType }?,
  attribute accumulate { p_ST_TLBehaviorAccumulateType }?,
  attribute xfrmType { p_ST_TLBehaviorTransformType }?,
  attribute from { xsd:string }?,
  attribute to { xsd:string }?,
  attribute by { xsd:string }?,
  attribute rctx { xsd:string }?,
  attribute override { p_ST_TLBehaviorOverrideType }?,
  element cTn { p_CT_TLCommonTimeNodeData },
  element tgtEl { p_CT_TLTimeTargetElement },
  element attrNameLst { p_CT_TLBehaviorAttributeNameList }?
p_CT_TLAnimVariantBooleanVal = attribute val { xsd:boolean }
p_CT_TLAnimVariantIntegerVal = attribute val { xsd:int }
p_CT_TLAnimVariantFloatVal = attribute val { xsd:float }
p_CT_TLAnimVariantStringVal = attribute val { xsd:string }
p_CT_TLAnimVariant =
  element boolVal { p_CT_TLAnimVariantBooleanVal }
  | element intVal { p_CT_TLAnimVariantIntegerVal }
  | element fltVal { p_CT_TLAnimVariantFloatVal }
  | element strVal { p_CT_TLAnimVariantStringVal }
  | element clrVal { a_CT_Color }
p_ST_TLTimeAnimateValueTime =
  a_ST_PositiveFixedPercentage | p_ST_TLTimeIndefinite
p_CT_TLTimeAnimateValue =

  ## default value: indefinite
  attribute tm { p_ST_TLTimeAnimateValueTime }?,
  attribute fmla { xsd:string }?,
  element val { p_CT_TLAnimVariant }?
p_CT_TLTimeAnimateValueList = element tav { p_CT_TLTimeAnimateValue }*
p_ST_TLAnimateBehaviorCalcMode = "discrete" | "lin" | "fmla"
p_ST_TLAnimateBehaviorValueType = "str" | "num" | "clr"
p_CT_TLAnimateBehavior =
  attribute by { xsd:string }?,
  attribute from { xsd:string }?,
  attribute to { xsd:string }?,
  attribute calcmode { p_ST_TLAnimateBehaviorCalcMode }?,
```



```
  attribute valueType { p_ST_TLAnimateBehaviorValueType }?,
  element cBhvr { p_CT_TLCommonBehaviorData },
  element tavLst { p_CT_TLTimeAnimateValueList }?
p_CT_TLByRgbColorTransform =
  attribute r { a_ST_FixedPercentage },
  attribute g { a_ST_FixedPercentage },
  attribute b { a_ST_FixedPercentage }
p_CT_TLByHslColorTransform =
  attribute h { a_ST_Angle },
  attribute s { a_ST_FixedPercentage },
  attribute l { a_ST_FixedPercentage }
p_CT_TLByAnimateColorTransform =
  element rgb { p_CT_TLByRgbColorTransform }
  | element hsl { p_CT_TLByHslColorTransform }
p_ST_TLAnimateColorSpace = "rgb" | "hsl"
p_ST_TLAnimateColorDirection = "cw" | "ccw"
p_CT_TLAnimateColorBehavior =
  attribute clrSpc { p_ST_TLAnimateColorSpace }?,
  attribute dir { p_ST_TLAnimateColorDirection }?,
  element cBhvr { p_CT_TLCommonBehaviorData },
  element by { p_CT_TLByAnimateColorTransform }?,
  element from { a_CT_Color }?,
  element to { a_CT_Color }?
p_ST_TLAnimateEffectTransition = "in" | "out" | "none"
p_CT_TLAnimateEffectBehavior =
  attribute transition { p_ST_TLAnimateEffectTransition }?,
  attribute filter { xsd:string }?,
  attribute prLst { xsd:string }?,
  element cBhvr { p_CT_TLCommonBehaviorData },
  element progress { p_CT_TLAnimVariant }?
p_ST_TLAnimateMotionBehaviorOrigin = "parent" | "layout"
p_ST_TLAnimateMotionPathEditMode = "relative" | "fixed"
p_CT_TLPoint =
  attribute x { a_ST_Percentage },
  attribute y { a_ST_Percentage }
p_CT_TLAnimateMotionBehavior =
  attribute origin { p_ST_TLAnimateMotionBehaviorOrigin }?,
  attribute path { xsd:string }?,
  attribute pathEditMode { p_ST_TLAnimateMotionPathEditMode }?,
  attribute rAng { a_ST_Angle }?,
  attribute ptsTypes { xsd:string }?,
  element cBhvr { p_CT_TLCommonBehaviorData },
  element by { p_CT_TLPoint }?,
  element from { p_CT_TLPoint }?,
  element to { p_CT_TLPoint }?,
  element rCtr { p_CT_TLPoint }?
p_CT_TLAnimateRotationBehavior =
  attribute by { a_ST_Angle }?,
  attribute from { a_ST_Angle }?,
  attribute to { a_ST_Angle }?,
  element cBhvr { p_CT_TLCommonBehaviorData }
p_CT_TLAnimateScaleBehavior =
  attribute zoomContents { xsd:boolean }?,
```

```
  element cBhvr { p_CT_TLCommonBehaviorData },
  element by { p_CT_TLPoint }?,
  element from { p_CT_TLPoint }?,
  element to { p_CT_TLPoint }?
p_ST_TLCommandType = "evt" | "call" | "verb"
p_CT_TLCommandBehavior =
  attribute type { p_ST_TLCommandType }?,
  attribute cmd { xsd:string }?,
  element cBhvr { p_CT_TLCommonBehaviorData }
p_CT_TLSetBehavior =
  element cBhvr { p_CT_TLCommonBehaviorData },
  element to { p_CT_TLAnimVariant }?
p_CT_TLCommonMediaNodeData =

  ## default value: 50%
  attribute vol { a_ST_PositiveFixedPercentage }?,

  ## default value: false
  attribute mute { xsd:boolean }?,

  ## default value: 1
  attribute numSld { xsd:unsignedInt }?,

  ## default value: true
  attribute showWhenStopped { xsd:boolean }?,
  element cTn { p_CT_TLCommonTimeNodeData },
  element tgtEl { p_CT_TLTimeTargetElement }
p_CT_TLMediaNodeAudio =

  ## default value: false
  attribute isNarration { xsd:boolean }?,
  element cMediaNode { p_CT_TLCommonMediaNodeData }
p_CT_TLMediaNodeVideo =

  ## default value: false
  attribute fullScrn { xsd:boolean }?,
  element cMediaNode { p_CT_TLCommonMediaNodeData }
p_AG_TLBuild =
  attribute spid { a_ST_DrawingElementId },
  attribute grpId { xsd:unsignedInt },

  ## default value: false
  attribute uiExpand { xsd:boolean }?
p_CT_TLTemplate =

  ## default value: 0
  attribute lvl { xsd:unsignedInt }?,
  element tnLst { p_CT_TimeNodeList }
p_CT_TLTemplateList = element tmpl { p_CT_TLTemplate }*
p_ST_TLParaBuildType = "allAtOnce" | "p" | "cust" | "whole"
p_CT_TLBuildParagraph =
  p_AG_TLBuild,

```



```
  ## default value: whole
  attribute build { p_ST_TLParaBuildType }?,

  ## default value: 1
  attribute bldLvl { xsd:unsignedInt }?,

  ## default value: false
  attribute animBg { xsd:boolean }?,

  ## default value: true
  attribute autoUpdateAnimBg { xsd:boolean }?,

  ## default value: false
  attribute rev { xsd:boolean }?,

  ## default value: indefinite
  attribute advAuto { p_ST_TLTime }?,
  element tmplLst { p_CT_TLTemplateList }?
p_ST_TLDiagramBuildType =
  "whole"
  | "depthByNode"
  | "depthByBranch"
  | "breadthByNode"
  | "breadthByLvl"
  | "cw"
  | "cwIn"
  | "cwOut"
  | "ccw"
  | "ccwIn"
  | "ccwOut"
  | "inByRing"
  | "outByRing"
  | "up"
  | "down"
  | "allAtOnce"
  | "cust"
p_CT_TLBuildDiagram =
  p_AG_TLBuild,

  ## default value: whole
  attribute bld { p_ST_TLDiagramBuildType }?
p_ST_TLOleChartBuildType =
  "allAtOnce" | "series" | "category" | "seriesEl" | "categoryEl"
p_CT_TLOleBuildChart =
  p_AG_TLBuild,

  ## default value: allAtOnce
  attribute bld { p_ST_TLOleChartBuildType }?,

  ## default value: true
  attribute animBg { xsd:boolean }?
p_CT_TLGraphicalObjectBuild =
  p_AG_TLBuild,
```

```
  (element bldAsOne { p_CT_Empty }
   | element bldSub { a_CT_AnimationGraphicalObjectBuildProperties })
p_CT_BuildList =
  (element bldP { p_CT_TLBuildParagraph }
   | element bldDgm { p_CT_TLBuildDiagram }
   | element bldOleChart { p_CT_TLOleBuildChart }
   | element bldGraphic { p_CT_TLGraphicalObjectBuild })+
p_CT_SlideTiming =
  element tnLst { p_CT_TimeNodeList }?,
  element bldLst { p_CT_BuildList }?,
  element extLst { p_CT_ExtensionListModify }?
p_CT_Empty = empty
p_ST_Name = xsd:string
p_ST_Direction = "horz" | "vert"
p_ST_Index = xsd:unsignedInt
p_CT_IndexRange =
  attribute st { p_ST_Index },
  attribute end { p_ST_Index }
p_CT_SlideRelationshipListEntry = r_id
p_CT_SlideRelationshipList =
  element sld { p_CT_SlideRelationshipListEntry }*
p_CT_CustomShowId = attribute id { xsd:unsignedInt }
p_EG_SlideListChoice =
  element sldAll { p_CT_Empty }
  | element sldRg { p_CT_IndexRange }
  | element custShow { p_CT_CustomShowId }
p_CT_CustomerData = r_id
p_CT_TagsData = r_id
p_CT_CustomerDataList =
  (element custData { p_CT_CustomerData }*,
   element tags { p_CT_TagsData }?)?
p_CT_Extension =
  attribute uri { xsd:token },
  p_CT_Extension_any*
p_CT_Extension_any =
  element * - (o:* | v:* | w10:* | x:*) {
    anyAttribute*,
    mixed { anyElement* }
  }
p_EG_ExtensionList = element ext { p_CT_Extension }*
p_CT_ExtensionList = p_EG_ExtensionList?
p_CT_ExtensionListModify =

  ## default value: false
  attribute mod { xsd:boolean }?,
  p_EG_ExtensionList?
p_CT_CommentAuthor =
  attribute id { xsd:unsignedInt },
  attribute name { p_ST_Name },
  attribute initials { p_ST_Name },
  attribute lastIdx { xsd:unsignedInt },
  attribute clrIdx { xsd:unsignedInt },
  element extLst { p_CT_ExtensionList }?
```



```
p_CT_CommentAuthorList = element cmAuthor { p_CT_CommentAuthor }*
p_cmAuthorLst = element cmAuthorLst { p_CT_CommentAuthorList }
p_CT_Comment =
  attribute authorId { xsd:unsignedInt },
  attribute dt { xsd:dateTime }?,
  attribute idx { p_ST_Index },
  element pos { a_CT_Point2D },
  element text { xsd:string },
  element extLst { p_CT_ExtensionListModify }?
p_CT_CommentList = element cm { p_CT_Comment }*
p_cmLst = element cmLst { p_CT_CommentList }
p_AG_Ole =
  attribute name { xsd:string }?,

  ## default value: false
  attribute showAsIcon { xsd:boolean }?,
  r_id?,
  attribute imgW { a_ST_PositiveCoordinate32 }?,
  attribute imgH { a_ST_PositiveCoordinate32 }?
p_ST_OleObjectFollowColorScheme = "none" | "full" | "textAndBackground"
p_CT_OleObjectEmbed =

  ## default value: none
  attribute followColorScheme { p_ST_OleObjectFollowColorScheme }?,
  element extLst { p_CT_ExtensionList }?
p_CT_OleObjectLink =

  ## default value: false
  attribute updateAutomatic { xsd:boolean }?,
  element extLst { p_CT_ExtensionList }?
p_CT_OleObject =
  p_AG_Ole,
  attribute progId { xsd:string }?,
  (element embed { p_CT_OleObjectEmbed }
   | element link { p_CT_OleObjectLink }
  ),
  (attribute spid { a_ST_ShapeID } | element pic { p_CT_Picture })
p_oleObj = element oleObj { p_CT_OleObject }
p_CT_Control =
  p_AG_Ole,
  element extLst { p_CT_ExtensionList }?,
  (attribute spid { a_ST_ShapeID } | element pic { p_CT_Picture })
p_CT_ControlList = element control { p_CT_Control }*
p_ST_SlideId =
  xsd:unsignedInt { minInclusive = "256" maxExclusive = "2147483648" }
p_CT_SlideIdListEntry =
  attribute id { p_ST_SlideId },
  r_id,
  element extLst { p_CT_ExtensionList }?
p_CT_SlideIdList = element sldId { p_CT_SlideIdListEntry }*
p_ST_SlideMasterId = xsd:unsignedInt { minInclusive = "2147483648" }
p_CT_SlideMasterIdListEntry =
  attribute id { p_ST_SlideMasterId }?,
```

```
  r_id,
  element extLst { p_CT_ExtensionList }?
p_CT_SlideMasterIdList =
  element sldMasterId { p_CT_SlideMasterIdListEntry }*
p_CT_NotesMasterIdListEntry =
  r_id,
  element extLst { p_CT_ExtensionList }?
p_CT_NotesMasterIdList =
  element notesMasterId { p_CT_NotesMasterIdListEntry }?
p_CT_HandoutMasterIdListEntry =
  r_id,
  element extLst { p_CT_ExtensionList }?
p_CT_HandoutMasterIdList =
  element handoutMasterId { p_CT_HandoutMasterIdListEntry }?
p_CT_EmbeddedFontDataId = r_id
p_CT_EmbeddedFontListEntry =
  element font { a_CT_TextFont },
  element regular { p_CT_EmbeddedFontDataId }?,
  element bold { p_CT_EmbeddedFontDataId }?,
  element italic { p_CT_EmbeddedFontDataId }?,
  element boldItalic { p_CT_EmbeddedFontDataId }?
p_CT_EmbeddedFontList =
  element embeddedFont { p_CT_EmbeddedFontListEntry }*
p_CT_SmartTags = r_id
p_CT_CustomShow =
  attribute name { p_ST_Name },
  attribute id { xsd:unsignedInt },
  element sldLst { p_CT_SlideRelationshipList },
  element extLst { p_CT_ExtensionList }?
p_CT_CustomShowList = element custShow { p_CT_CustomShow }*
p_ST_PhotoAlbumLayout =
  "fitToSlide"
  | "1pic"
  | "2pic"
  | "4pic"
  | "1picTitle"
  | "2picTitle"
  | "4picTitle"
p_ST_PhotoAlbumFrameShape =
  "frameStyle1"
  | "frameStyle2"
  | "frameStyle3"
  | "frameStyle4"
  | "frameStyle5"
  | "frameStyle6"
  | "frameStyle7"
p_CT_PhotoAlbum =

  ## default value: false
  attribute bw { xsd:boolean }?,

  ## default value: false
  attribute showCaptions { xsd:boolean }?,
```



```
  ## default value: fitToSlide
  attribute layout { p_ST_PhotoAlbumLayout }?,

  ## default value: frameStyle1
  attribute frame { p_ST_PhotoAlbumFrameShape }?,
  element extLst { p_CT_ExtensionList }?
p_ST_SlideSizeCoordinate =
  xsd:int {
    minInclusive = "914400"
    maxInclusive = "51206400"
  }
p_ST_SlideSizeType =
  "screen4x3"
  | "letter"
  | "A4"
  | "35mm"
  | "overhead"
  | "banner"
  | "custom"
  | "ledger"
  | "A3"
  | "B4ISO"
  | "B5ISO"
  | "B4JIS"
  | "B5JIS"
  | "hagakiCard"
  | "screen16x9"
  | "screen16x10"
p_CT_SlideSize =
  attribute cx { p_ST_SlideSizeCoordinate },
  attribute cy { p_ST_SlideSizeCoordinate },

  ## default value: custom
  attribute type { p_ST_SlideSizeType }?
p_CT_Kinsoku =
  attribute lang { xsd:string }?,
  attribute invalStChars { xsd:string },
  attribute invalEndChars { xsd:string }
p_ST_BookmarkIdSeed =
  xsd:unsignedInt { minInclusive = "1" maxExclusive = "2147483648" }
p_CT_ModifyVerifier =
  attribute algorithmName { xsd:string }?,
  attribute hashValue { xsd:base64Binary }?,
  attribute saltValue { xsd:base64Binary }?,
  attribute spinValue { xsd:unsignedInt }?,
  attribute cryptProviderType { s_ST_CryptProv }?,
  attribute cryptAlgorithmClass { s_ST_AlgClass }?,
  attribute cryptAlgorithmType { s_ST_AlgType }?,
  attribute cryptAlgorithmSid { xsd:unsignedInt }?,
  attribute spinCount { xsd:unsignedInt }?,
  attribute saltData { xsd:base64Binary }?,
  attribute hashData { xsd:base64Binary }?,
```

```
  attribute cryptProvider { xsd:string }?,
  attribute algIdExt { xsd:unsignedInt }?,
  attribute algIdExtSource { xsd:string }?,
  attribute cryptProviderTypeExt { xsd:unsignedInt }?,
  attribute cryptProviderTypeExtSource { xsd:string }?
p_CT_Presentation =

  ## default value: 50%
  attribute serverZoom { a_ST_Percentage }?,

  ## default value: 1
  attribute firstSlideNum { xsd:int }?,

  ## default value: true
  attribute showSpecialPlsOnTitleSld { xsd:boolean }?,

  ## default value: false
  attribute rtl { xsd:boolean }?,

  ## default value: false
  attribute removePersonalInfoOnSave { xsd:boolean }?,

  ## default value: false
  attribute compatMode { xsd:boolean }?,

  ## default value: true
  attribute strictFirstAndLastChars { xsd:boolean }?,

  ## default value: false
  attribute embedTrueTypeFonts { xsd:boolean }?,

  ## default value: false
  attribute saveSubsetFonts { xsd:boolean }?,

  ## default value: true
  attribute autoCompressPictures { xsd:boolean }?,

  ## default value: 1
  attribute bookmarkIdSeed { p_ST_BookmarkIdSeed }?,
  attribute conformance { s_ST_ConformanceClass }?,
  element sldMasterIdLst { p_CT_SlideMasterIdList }?,
  element notesMasterIdLst { p_CT_NotesMasterIdList }?,
  element handoutMasterIdLst { p_CT_HandoutMasterIdList }?,
  element sldIdLst { p_CT_SlideIdList }?,
  element sldSz { p_CT_SlideSize }?,
  element notesSz { a_CT_PositiveSize2D },
  element smartTags { p_CT_SmartTags }?,
  element embeddedFontLst { p_CT_EmbeddedFontList }?,
  element custShowLst { p_CT_CustomShowList }?,
  element photoAlbum { p_CT_PhotoAlbum }?,
  element custDataLst { p_CT_CustomerDataList }?,
  element kinsoku { p_CT_Kinsoku }?,
  element defaultTextStyle { a_CT_TextListStyle }?,
```



```
  element modifyVerifier { p_CT_ModifyVerifier }?,
  element extLst { p_CT_ExtensionList }?
p_presentation = element presentation { p_CT_Presentation }
p_CT_HtmlPublishProperties =

  ## default value: true
  attribute showSpeakerNotes { xsd:boolean }?,
  attribute target { xsd:string }?,
  attribute title { xsd:string }?,
  r_id,
  p_EG_SlideListChoice,
  element extLst { p_CT_ExtensionList }?
p_ST_WebColorType =
  "none"
  | "browser"
  | "presentationText"
  | "presentationAccent"
  | "whiteTextOnBlack"
  | "blackTextOnWhite"
p_ST_WebScreenSize =
  "544x376"
  | "640x480"
  | "720x512"
  | "800x600"
  | "1024x768"
  | "1152x882"
  | "1152x900"
  | "1280x1024"
  | "1600x1200"
  | "1800x1400"
  | "1920x1200"
p_ST_WebEncoding = xsd:string
p_CT_WebProperties =

  ## default value: false
  attribute showAnimation { xsd:boolean }?,

  ## default value: true
  attribute resizeGraphics { xsd:boolean }?,

  ## default value: false
  attribute allowPng { xsd:boolean }?,

  ## default value: false
  attribute relyOnVml { xsd:boolean }?,

  ## default value: true
  attribute organizeInFolders { xsd:boolean }?,

  ## default value: true
  attribute useLongFilenames { xsd:boolean }?,

  ## default value: 800x600
```

```
  attribute imgSz { p_ST_WebScreenSize }?,
  attribute encoding { p_ST_WebEncoding }?,

  ## default value: whiteTextOnBlack
  attribute clr { p_ST_WebColorType }?,
  element extLst { p_CT_ExtensionList }?
p_ST_PrintWhat =
  "slides"
  | "handouts1"
  | "handouts2"
  | "handouts3"
  | "handouts4"
  | "handouts6"
  | "handouts9"
  | "notes"
  | "outline"
p_ST_PrintColorMode = "bw" | "gray" | "clr"
p_CT_PrintProperties =

  ## default value: slides
  attribute prnWhat { p_ST_PrintWhat }?,

  ## default value: clr
  attribute clrMode { p_ST_PrintColorMode }?,

  ## default value: false
  attribute hiddenSlides { xsd:boolean }?,

  ## default value: false
  attribute scaleToFitPaper { xsd:boolean }?,

  ## default value: false
  attribute frameSlides { xsd:boolean }?,
  element extLst { p_CT_ExtensionList }?
p_CT_ShowInfoBrowse =

  ## default value: true
  attribute showScrollbar { xsd:boolean }?
p_CT_ShowInfoKiosk =

  ## default value: 300000
  attribute restart { xsd:unsignedInt }?
p_EG_ShowType =
  element present { p_CT_Empty }
  | element browse { p_CT_ShowInfoBrowse }
  | element kiosk { p_CT_ShowInfoKiosk }
p_CT_ShowProperties =

  ## default value: false
  attribute loop { xsd:boolean }?,

  ## default value: false
  attribute showNarration { xsd:boolean }?,
```



```
  ## default value: true
  attribute showAnimation { xsd:boolean }?,

  ## default value: true
  attribute useTimings { xsd:boolean }?,
  (p_EG_ShowType?,
   p_EG_SlideListChoice?,
   element penClr { a_CT_Color }?,
   element extLst { p_CT_ExtensionList }?)?
p_CT_PresentationProperties =
  element htmlPubPr { p_CT_HtmlPublishProperties }?,
  element webPr { p_CT_WebProperties }?,
  element prnPr { p_CT_PrintProperties }?,
  element showPr { p_CT_ShowProperties }?,
  element clrMru { a_CT_ColorMRU }?,
  element extLst { p_CT_ExtensionList }?
p_presentationPr =
  element presentationPr { p_CT_PresentationProperties }
p_CT_HeaderFooter =

  ## default value: true
  attribute sldNum { xsd:boolean }?,

  ## default value: true
  attribute hdr { xsd:boolean }?,

  ## default value: true
  attribute ftr { xsd:boolean }?,

  ## default value: true
  attribute dt { xsd:boolean }?,
  element extLst { p_CT_ExtensionListModify }?
p_ST_PlaceholderType =
  "title"
  | "body"
  | "ctrTitle"
  | "subTitle"
  | "dt"
  | "sldNum"
  | "ftr"
  | "hdr"
  | "obj"
  | "chart"
  | "tbl"
  | "clipArt"
  | "dgm"
  | "media"
  | "sldImg"
  | "pic"
p_ST_PlaceholderSize = "full" | "half" | "quarter"
p_CT_Placeholder =

```

```
  ## default value: obj
  attribute type { p_ST_PlaceholderType }?,

  ## default value: horz
  attribute orient { p_ST_Direction }?,

  ## default value: full
  attribute sz { p_ST_PlaceholderSize }?,

  ## default value: 0
  attribute idx { xsd:unsignedInt }?,

  ## default value: false
  attribute hasCustomPrompt { xsd:boolean }?,
  element extLst { p_CT_ExtensionListModify }?
p_CT_ApplicationNonVisualDrawingProps =

  ## default value: false
  attribute isPhoto { xsd:boolean }?,

  ## default value: false
  attribute userDrawn { xsd:boolean }?,
  element ph { p_CT_Placeholder }?,
  a_EG_Media?,
  element custDataLst { p_CT_CustomerDataList }?,
  element extLst { p_CT_ExtensionList }?
p_CT_ShapeNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvSpPr { a_CT_NonVisualDrawingShapeProps },
  element nvPr { p_CT_ApplicationNonVisualDrawingProps }
p_CT_Shape =

  ## default value: false
  attribute useBgFill { xsd:boolean }?,
  element nvSpPr { p_CT_ShapeNonVisual },
  element spPr { a_CT_ShapeProperties },
  element style { a_CT_ShapeStyle }?,
  element txBody { a_CT_TextBody }?,
  element extLst { p_CT_ExtensionListModify }?
p_CT_ConnectorNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvCxnSpPr { a_CT_NonVisualConnectorProperties },
  element nvPr { p_CT_ApplicationNonVisualDrawingProps }
p_CT_Connector =
  element nvCxnSpPr { p_CT_ConnectorNonVisual },
  element spPr { a_CT_ShapeProperties },
  element style { a_CT_ShapeStyle }?,
  element extLst { p_CT_ExtensionListModify }?
p_CT_PictureNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvPicPr { a_CT_NonVisualPictureProperties },
  element nvPr { p_CT_ApplicationNonVisualDrawingProps }
p_CT_Picture =
```



```
  element nvPicPr { p_CT_PictureNonVisual },
  element blipFill { a_CT_BlipFillProperties },
  element spPr { a_CT_ShapeProperties },
  element style { a_CT_ShapeStyle }?,
  element extLst { p_CT_ExtensionListModify }?
p_CT_GraphicalObjectFrameNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvGraphicFramePr { a_CT_NonVisualGraphicFrameProperties },
  element nvPr { p_CT_ApplicationNonVisualDrawingProps }
p_CT_GraphicalObjectFrame =
  attribute bwMode { a_ST_BlackWhiteMode }?,
  element nvGraphicFramePr { p_CT_GraphicalObjectFrameNonVisual },
  element xfrm { a_CT_Transform2D },
  a_graphic,
  element extLst { p_CT_ExtensionListModify }?
p_CT_GroupShapeNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvGrpSpPr { a_CT_NonVisualGroupDrawingShapeProps },
  element nvPr { p_CT_ApplicationNonVisualDrawingProps }
p_CT_GroupShape =
  element nvGrpSpPr { p_CT_GroupShapeNonVisual },
  element grpSpPr { a_CT_GroupShapeProperties },
  (element sp { p_CT_Shape }
   | element grpSp { p_CT_GroupShape }
   | element graphicFrame { p_CT_GraphicalObjectFrame }
   | element cxnSp { p_CT_Connector }
   | element pic { p_CT_Picture }
   | element contentPart { p_CT_Rel })*,
  element extLst { p_CT_ExtensionListModify }?
p_CT_Rel = r_id
p_EG_TopLevelSlide = element clrMap { a_CT_ColorMapping }
p_EG_ChildSlide = element clrMapOvr { a_CT_ColorMappingOverride }?
p_AG_ChildSlide =

  ## default value: true
  attribute showMasterSp { xsd:boolean }?,

  ## default value: true
  attribute showMasterPhAnim { xsd:boolean }?
p_CT_BackgroundProperties =

  ## default value: false
  attribute shadeToTitle { xsd:boolean }?,
  a_EG_FillProperties,
  a_EG_EffectProperties?,
  element extLst { p_CT_ExtensionList }?
p_EG_Background =
  element bgPr { p_CT_BackgroundProperties }
  | element bgRef { a_CT_StyleMatrixReference }
p_CT_Background =

  ## default value: white
  attribute bwMode { a_ST_BlackWhiteMode }?,
```

```
  p_EG_Background
p_CT_CommonSlideData =
  attribute name { xsd:string }?,
  element bg { p_CT_Background }?,
  element spTree { p_CT_GroupShape },
  element custDataLst { p_CT_CustomerDataList }?,
  element controls { p_CT_ControlList }?,
  element extLst { p_CT_ExtensionList }?
p_CT_Slide =
  p_AG_ChildSlide,

  ## default value: true
  attribute show { xsd:boolean }?,
  element cSld { p_CT_CommonSlideData },
  p_EG_ChildSlide?,
  element transition { p_CT_SlideTransition }?,
  element timing { p_CT_SlideTiming }?,
  element extLst { p_CT_ExtensionListModify }?
p_sld = element sld { p_CT_Slide }
p_ST_SlideLayoutType =
  "title"
  | "tx"
  | "twoColTx"
  | "tbl"
  | "txAndChart"
  | "chartAndTx"
  | "dgm"
  | "chart"
  | "txAndClipArt"
  | "clipArtAndTx"
  | "titleOnly"
  | "blank"
  | "txAndObj"
  | "objAndTx"
  | "objOnly"
  | "obj"
  | "txAndMedia"
  | "mediaAndTx"
  | "objOverTx"
  | "txOverObj"
  | "txAndTwoObj"
  | "twoObjAndTx"
  | "twoObjOverTx"
  | "fourObj"
  | "vertTx"
  | "clipArtAndVertTx"
  | "vertTitleAndTx"
  | "vertTitleAndTxOverChart"
  | "twoObj"
  | "objAndTwoObj"
  | "twoObjAndObj"
  | "cust"
  | "secHead"
```



```
  | "twoTxTwoObj"
  | "objTx"
  | "picTx"
p_CT_SlideLayout =
  p_AG_ChildSlide,
  attribute matchingName { xsd:string }?,

  ## default value: cust
  attribute type { p_ST_SlideLayoutType }?,

  ## default value: false
  attribute preserve { xsd:boolean }?,

  ## default value: false
  attribute userDrawn { xsd:boolean }?,
  element cSld { p_CT_CommonSlideData },
  p_EG_ChildSlide?,
  element transition { p_CT_SlideTransition }?,
  element timing { p_CT_SlideTiming }?,
  element hf { p_CT_HeaderFooter }?,
  element extLst { p_CT_ExtensionListModify }?
p_sldLayout = element sldLayout { p_CT_SlideLayout }
p_CT_SlideMasterTextStyles =
  element titleStyle { a_CT_TextListStyle }?,
  element bodyStyle { a_CT_TextListStyle }?,
  element otherStyle { a_CT_TextListStyle }?,
  element extLst { p_CT_ExtensionList }?
p_ST_SlideLayoutId = xsd:unsignedInt { minInclusive = "2147483648" }
p_CT_SlideLayoutIdListEntry =
  attribute id { p_ST_SlideLayoutId }?,
  r_id,
  element extLst { p_CT_ExtensionList }?
p_CT_SlideLayoutIdList =
  element sldLayoutId { p_CT_SlideLayoutIdListEntry }*
p_CT_SlideMaster =

  ## default value: false
  attribute preserve { xsd:boolean }?,
  element cSld { p_CT_CommonSlideData },
  p_EG_TopLevelSlide,
  element sldLayoutIdLst { p_CT_SlideLayoutIdList }?,
  element transition { p_CT_SlideTransition }?,
  element timing { p_CT_SlideTiming }?,
  element hf { p_CT_HeaderFooter }?,
  element txStyles { p_CT_SlideMasterTextStyles }?,
  element extLst { p_CT_ExtensionListModify }?
p_sldMaster = element sldMaster { p_CT_SlideMaster }
p_CT_HandoutMaster =
  element cSld { p_CT_CommonSlideData },
  p_EG_TopLevelSlide,
  element hf { p_CT_HeaderFooter }?,
  element extLst { p_CT_ExtensionListModify }?
p_handoutMaster = element handoutMaster { p_CT_HandoutMaster }
```

```
p_CT_NotesMaster =
  element cSld { p_CT_CommonSlideData },
  p_EG_TopLevelSlide,
  element hf { p_CT_HeaderFooter }?,
  element notesStyle { a_CT_TextListStyle }?,
  element extLst { p_CT_ExtensionListModify }?
p_notesMaster = element notesMaster { p_CT_NotesMaster }
p_CT_NotesSlide =
  p_AG_ChildSlide,
  element cSld { p_CT_CommonSlideData },
  p_EG_ChildSlide?,
  element extLst { p_CT_ExtensionListModify }?
p_notes = element notes { p_CT_NotesSlide }
p_CT_SlideSyncProperties =
  attribute serverSldId { xsd:string },
  attribute serverSldModifiedTime { xsd:dateTime },
  attribute clientInsertedTime { xsd:dateTime },
  element extLst { p_CT_ExtensionList }?
p_sldSyncPr = element sldSyncPr { p_CT_SlideSyncProperties }
p_CT_StringTag =
  attribute name { xsd:string },
  attribute val { xsd:string }
p_CT_TagList = element tag { p_CT_StringTag }*
p_tagLst = element tagLst { p_CT_TagList }
p_ST_SplitterBarState = "minimized" | "restored" | "maximized"
p_ST_ViewType =
  "sldView"
  | "sldMasterView"
  | "notesView"
  | "handoutView"
  | "notesMasterView"
  | "outlineView"
  | "sldSorterView"
  | "sldThumbnailView"
p_CT_NormalViewPortion =
  attribute sz { a_ST_PositiveFixedPercentage },

  ## default value: true
  attribute autoAdjust { xsd:boolean }?
p_CT_NormalViewProperties =

  ## default value: true
  attribute showOutlineIcons { xsd:boolean }?,

  ## default value: false
  attribute snapVertSplitter { xsd:boolean }?,

  ## default value: restored
  attribute vertBarState { p_ST_SplitterBarState }?,

  ## default value: restored
  attribute horzBarState { p_ST_SplitterBarState }?,

```



```
  ## default value: false
  attribute preferSingleView { xsd:boolean }?,
  element restoredLeft { p_CT_NormalViewPortion },
  element restoredTop { p_CT_NormalViewPortion },
  element extLst { p_CT_ExtensionList }?
p_CT_CommonViewProperties =

  ## default value: false
  attribute varScale { xsd:boolean }?,
  element scale { a_CT_Scale2D },
  element origin { a_CT_Point2D }
p_CT_NotesTextViewProperties =
  element cViewPr { p_CT_CommonViewProperties },
  element extLst { p_CT_ExtensionList }?
p_CT_OutlineViewSlideEntry =
  r_id,

  ## default value: false
  attribute collapse { xsd:boolean }?
p_CT_OutlineViewSlideList = element sld { p_CT_OutlineViewSlideEntry }*
p_CT_OutlineViewProperties =
  element cViewPr { p_CT_CommonViewProperties },
  element sldLst { p_CT_OutlineViewSlideList }?,
  element extLst { p_CT_ExtensionList }?
p_CT_SlideSorterViewProperties =

  ## default value: true
  attribute showFormatting { xsd:boolean }?,
  element cViewPr { p_CT_CommonViewProperties },
  element extLst { p_CT_ExtensionList }?
p_CT_Guide =

  ## default value: vert
  attribute orient { p_ST_Direction }?,

  ## default value: 0
  attribute pos { a_ST_Coordinate32 }?
p_CT_GuideList = element guide { p_CT_Guide }*
p_CT_CommonSlideViewProperties =

  ## default value: true
  attribute snapToGrid { xsd:boolean }?,

  ## default value: false
  attribute snapToObjects { xsd:boolean }?,

  ## default value: false
  attribute showGuides { xsd:boolean }?,
  element cViewPr { p_CT_CommonViewProperties },
  element guideLst { p_CT_GuideList }?
p_CT_SlideViewProperties =
  element cSldViewPr { p_CT_CommonSlideViewProperties },
  element extLst { p_CT_ExtensionList }?
```

```
p_CT_NotesViewProperties =
  element cSldViewPr { p_CT_CommonSlideViewProperties },
  element extLst { p_CT_ExtensionList }?
p_CT_ViewProperties =

  ## default value: sldView
  attribute lastView { p_ST_ViewType }?,

  ## default value: true
  attribute showComments { xsd:boolean }?,
  (element normalViewPr { p_CT_NormalViewProperties }?,
   element slideViewPr { p_CT_SlideViewProperties }?,
   element outlineViewPr { p_CT_OutlineViewProperties }?,
   element notesTextViewPr { p_CT_NotesTextViewProperties }?,
   element sorterViewPr { p_CT_SlideSorterViewProperties }?,
   element notesViewPr { p_CT_NotesViewProperties }?,
   element gridSpacing { a_CT_PositiveSize2D }?,
   element extLst { p_CT_ExtensionList }?)?
p_viewPr = element viewPr { p_CT_ViewProperties }
```

B.3.1 Part Schemas

B.3.1.1 Comment Authors Part

This schema is available in the file PresentationML_Comment_Authors.rnc.

```
include "pml.rnc"
include "shared-relationshipReference.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = p_cmAuthorLst
```

B.3.1.2 Comments Part

This schema is available in the file PresentationML_Comments.rnc.

```
include "pml.rnc"
include "shared-relationshipReference.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = p_cmLst
```


B.3.1.3 Handout Master Part

This schema is available in the file PresentationML_Handout_Master.rnc.

```
include "pml.rnc"
include "shared-relationshipReference.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = p_handoutMaster
```

B.3.1.4 Notes Master Part

This schema is available in the file PresentationML_Notes_Master.rnc.

```
include "pml.rnc"
include "shared-relationshipReference.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = p_notesMaster
```

B.3.1.5 Notes Slide Part

This schema is available in the file PresentationML_Notes_Slide.rnc.

```
include "pml.rnc"
include "shared-relationshipReference.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = p_notes
```

B.3.1.6 Presentation Part

This schema is available in the file PresentationML_Presentation.rnc.

```
include "pml.rnc"
include "shared-relationshipReference.rnc"
```

```
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = p_presentation
```

B.3.1.7 Presentation Properties Part

This schema is available in the file PresentationML_Presentation_Properties.rnc.

```
include "pml.rnc"
include "shared-relationshipReference.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = p_presentationPr
```

B.3.1.8 Slide Part

This schema is available in the file PresentationML_Slide.rnc.

```
include "pml.rnc"
include "shared-relationshipReference.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = p_sld
```

B.3.1.9 Slide Layout Part

This schema is available in the file PresentationML_Slide_Layout.rnc.

```
include "pml.rnc"
include "shared-relationshipReference.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
```



```
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = p_sldLayout
```

B.3.1.10 Slide Master Part

This schema is available in the file PresentationML_Slide_Master.rnc.

```
include "pml.rnc"
include "shared-relationshipReference.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = p_sldMaster
```

B.3.1.11 Slide Synchronization Data Part

This schema is available in the file PresentationML_Slide_Synchronization_Data.rnc.

```
include "pml.rnc"
include "shared-relationshipReference.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = p_sldSyncPr
```

B.3.1.12 User Defined Tags Part

This schema is available in the file PresentationML_User-Defined_Tags.rnc.

```
include "pml.rnc"
include "shared-relationshipReference.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = p_tagLst
```

B.3.1.13 View Properties Part

This schema is available in the file PresentationML_View_Properties.rnc.

```
include "pml.rnc"
include "shared-relationshipReference.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = p_viewPr
```

B.4 DrawingML - Framework

B.4.1 DrawingML - Main

This schema is available in the file dml-main.rnc.

```
default namespace =
  "http://schemas.openxmlformats.org/drawingml/2006/main"
namespace a = "http://schemas.openxmlformats.org/drawingml/2006/main"
namespace o = "urn:schemas-microsoft-com:office:office"
namespace r =
  "http://schemas.openxmlformats.org/officeDocument/2006/relationships"
namespace s =
  "http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
namespace v = "urn:schemas-microsoft-com:vml"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"

a_CT_AudioFile =
  r_link,
  attribute contentType { xsd:string }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_VideoFile =
  r_link,
  attribute contentType { xsd:string }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_QuickTimeFile =
  r_link,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_AudioCDTime =
  attribute track { xsd:unsignedByte },

  ## default value: 0
  attribute time { xsd:unsignedInt }?
a_CT_AudioCD =
  element st { a_CT_AudioCDTime },
  element end { a_CT_AudioCDTime },
```



```
  element extLst { a_CT_OfficeArtExtensionList }?
a_EG_Media =
  element audioCd { a_CT_AudioCD }
  | element wavAudioFile { a_CT_EmbeddedWAVAudioFile }
  | element audioFile { a_CT_AudioFile }
  | element videoFile { a_CT_VideoFile }
  | element quickTimeFile { a_CT_QuickTimeFile }
a_videoFile = element videoFile { a_CT_VideoFile }
a_ST_StyleMatrixColumnIndex = xsd:unsignedInt
a_ST_FontCollectionIndex = "major" | "minor" | "none"
a_ST_ColorSchemeIndex =
  "dk1"
  | "lt1"
  | "dk2"
  | "lt2"
  | "accent1"
  | "accent2"
  | "accent3"
  | "accent4"
  | "accent5"
  | "accent6"
  | "hlink"
  | "folHlink"
a_CT_ColorScheme =
  attribute name { xsd:string },
  element dk1 { a_CT_Color },
  element lt1 { a_CT_Color },
  element dk2 { a_CT_Color },
  element lt2 { a_CT_Color },
  element accent1 { a_CT_Color },
  element accent2 { a_CT_Color },
  element accent3 { a_CT_Color },
  element accent4 { a_CT_Color },
  element accent5 { a_CT_Color },
  element accent6 { a_CT_Color },
  element hlink { a_CT_Color },
  element folHlink { a_CT_Color },
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_CustomColor =
  attribute name { xsd:string }?,
  a_EG_ColorChoice
a_CT_SupplementalFont =
  attribute script { xsd:string },
  attribute typeface { a_ST_TextTypeface }
a_CT_CustomColorList = element custClr { a_CT_CustomColor }*
a_CT_FontCollection =
  element latin { a_CT_TextFont },
  element ea { a_CT_TextFont },
  element cs { a_CT_TextFont },
  element font { a_CT_SupplementalFont }*,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_EffectStyleItem =
  a_EG_EffectProperties,
```

```
  element scene3d { a_CT_Scene3D }?,
  element sp3d { a_CT_Shape3D }?
a_CT_FontScheme =
  attribute name { xsd:string },
  element majorFont { a_CT_FontCollection },
  element minorFont { a_CT_FontCollection },
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_FillStyleList = a_EG_FillProperties+
a_CT_LineStyleList = element ln { a_CT_LineProperties }+
a_CT_EffectStyleList = element effectStyle { a_CT_EffectStyleItem }+
a_CT_BackgroundFillStyleList = a_EG_FillProperties+
a_CT_StyleMatrix =
  attribute name { xsd:string }?,
  element fillStyleLst { a_CT_FillStyleList },
  element lnStyleLst { a_CT_LineStyleList },
  element effectStyleLst { a_CT_EffectStyleList },
  element bgFillStyleLst { a_CT_BackgroundFillStyleList }
a_CT_BaseStyles =
  element clrScheme { a_CT_ColorScheme },
  element fontScheme { a_CT_FontScheme },
  element fmtScheme { a_CT_StyleMatrix },
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_OfficeArtExtension =
  attribute uri { xsd:token },
  a_CT_OfficeArtExtension_any*
a_CT_OfficeArtExtension_any =
  element * - (o:* | v:* | w10:* | x:*) {
    anyAttribute*,
    mixed { anyElement* }
  }
a_ST_Coordinate = a_ST_CoordinateUnqualified | s_ST_UniversalMeasure
a_ST_CoordinateUnqualified =
  xsd:long {
    minInclusive = "-27273042329600"
    maxInclusive = "27273042316900"
  }
a_ST_Coordinate32 = a_ST_Coordinate32Unqualified | s_ST_UniversalMeasure
a_ST_Coordinate32Unqualified = xsd:int
a_ST_PositiveCoordinate =
  xsd:long { minInclusive = "0" maxInclusive = "27273042316900" }
a_ST_PositiveCoordinate32 = xsd:int { minInclusive = "0" }
a_ST_Angle = xsd:int
a_CT_Angle = attribute val { a_ST_Angle }
a_ST_FixedAngle =
  xsd:int { minExclusive = "-5400000" maxExclusive = "5400000" }
a_ST_PositiveFixedAngle =
  xsd:int { minInclusive = "0" maxExclusive = "21600000" }
a_CT_PositiveFixedAngle = attribute val { a_ST_PositiveFixedAngle }
a_ST_Percentage = a_ST_PercentageDecimal | s_ST_Percentage
a_ST_PercentageDecimal = xsd:int
a_CT_Percentage = attribute val { a_ST_Percentage }
a_ST_PositivePercentage =
  a_ST_PositivePercentageDecimal | s_ST_PositivePercentage
```



```
a_ST_PositivePercentageDecimal = xsd:int { minInclusive = "0" }
a_CT_PositivePercentage = attribute val { a_ST_PositivePercentage }
a_ST_FixedPercentage =
  a_ST_FixedPercentageDecimal | s_ST_FixedPercentage
a_ST_FixedPercentageDecimal =
  xsd:int { minInclusive = "-100000" maxInclusive = "100000" }
a_CT_FixedPercentage = attribute val { a_ST_FixedPercentage }
a_ST_PositiveFixedPercentage =
  a_ST_PositiveFixedPercentageDecimal | s_ST_PositiveFixedPercentage
a_ST_PositiveFixedPercentageDecimal =
  xsd:int { minInclusive = "0" maxInclusive = "100000" }
a_CT_PositiveFixedPercentage =
  attribute val { a_ST_PositiveFixedPercentage }
a_CT_Ratio =
  attribute n { xsd:long },
  attribute d { xsd:long }
a_CT_Point2D =
  attribute x { a_ST_Coordinate },
  attribute y { a_ST_Coordinate }
a_CT_PositiveSize2D =
  attribute cx { a_ST_PositiveCoordinate },
  attribute cy { a_ST_PositiveCoordinate }
a_CT_ComplementTransform = empty
a_CT_InverseTransform = empty
a_CT_GrayscaleTransform = empty
a_CT_GammaTransform = empty
a_CT_InverseGammaTransform = empty
a_EG_ColorTransform =
  element tint { a_CT_PositiveFixedPercentage }
  | element shade { a_CT_PositiveFixedPercentage }
  | element comp { a_CT_ComplementTransform }
  | element inv { a_CT_InverseTransform }
  | element gray { a_CT_GrayscaleTransform }
  | element alpha { a_CT_PositiveFixedPercentage }
  | element alphaOff { a_CT_FixedPercentage }
  | element alphaMod { a_CT_PositivePercentage }
  | element hue { a_CT_PositiveFixedAngle }
  | element hueOff { a_CT_Angle }
  | element hueMod { a_CT_PositivePercentage }
  | element sat { a_CT_Percentage }
  | element satOff { a_CT_Percentage }
  | element satMod { a_CT_Percentage }
  | element lum { a_CT_Percentage }
  | element lumOff { a_CT_Percentage }
  | element lumMod { a_CT_Percentage }
  | element red { a_CT_Percentage }
  | element redOff { a_CT_Percentage }
  | element redMod { a_CT_Percentage }
  | element green { a_CT_Percentage }
  | element greenOff { a_CT_Percentage }
  | element greenMod { a_CT_Percentage }
  | element blue { a_CT_Percentage }
  | element blueOff { a_CT_Percentage }
```

```
  | element blueMod { a_CT_Percentage }
  | element gamma { a_CT_GammaTransform }
  | element invGamma { a_CT_InverseGammaTransform }
a_CT_ScRgbColor =
  attribute r { a_ST_Percentage },
  attribute g { a_ST_Percentage },
  attribute b { a_ST_Percentage },
  a_EG_ColorTransform*
a_CT_SRgbColor =
  attribute val { s_ST_HexColorRGB },
  a_EG_ColorTransform*
a_CT_HslColor =
  attribute hue { a_ST_PositiveFixedAngle },
  attribute sat { a_ST_Percentage },
  attribute lum { a_ST_Percentage },
  a_EG_ColorTransform*
a_ST_SystemColorVal =
  "scrollBar"
  | "background"
  | "activeCaption"
  | "inactiveCaption"
  | "menu"
  | "window"
  | "windowFrame"
  | "menuText"
  | "windowText"
  | "captionText"
  | "activeBorder"
  | "inactiveBorder"
  | "appWorkspace"
  | "highlight"
  | "highlightText"
  | "btnFace"
  | "btnShadow"
  | "grayText"
  | "btnText"
  | "inactiveCaptionText"
  | "btnHighlight"
  | "3dDkShadow"
  | "3dLight"
  | "infoText"
  | "infoBk"
  | "hotLight"
  | "gradientActiveCaption"
  | "gradientInactiveCaption"
  | "menuHighlight"
  | "menuBar"
a_CT_SystemColor =
  attribute val { a_ST_SystemColorVal },
  attribute lastClr { s_ST_HexColorRGB }?,
  a_EG_ColorTransform*
a_ST_SchemeColorVal =
  "bg1"
```



```
  | "tx1"
  | "bg2"
  | "tx2"
  | "accent1"
  | "accent2"
  | "accent3"
  | "accent4"
  | "accent5"
  | "accent6"
  | "hlink"
  | "folHlink"
  | "phClr"
  | "dk1"
  | "lt1"
  | "dk2"
  | "lt2"
a_CT_SchemeColor =
  attribute val { a_ST_SchemeColorVal },
  a_EG_ColorTransform*
a_ST_PresetColorVal =
  "aliceBlue"
  | "antiqueWhite"
  | "aqua"
  | "aquamarine"
  | "azure"
  | "beige"
  | "bisque"
  | "black"
  | "blanchedAlmond"
  | "blue"
  | "blueViolet"
  | "brown"
  | "burlyWood"
  | "cadetBlue"
  | "chartreuse"
  | "chocolate"
  | "coral"
  | "cornflowerBlue"
  | "cornsilk"
  | "crimson"
  | "cyan"
  | "darkBlue"
  | "darkCyan"
  | "darkGoldenrod"
  | "darkGray"
  | "darkGrey"
  | "darkGreen"
  | "darkKhaki"
  | "darkMagenta"
  | "darkOliveGreen"
  | "darkOrange"
  | "darkOrchid"
  | "darkRed"
```

```
    | "darkSalmon"
    | "darkSeaGreen"
    | "darkSlateBlue"
    | "darkSlateGray"
    | "darkSlateGrey"
    | "darkTurquoise"
    | "darkViolet"
    | "dkBlue"
    | "dkCyan"
    | "dkGoldenrod"
    | "dkGray"
    | "dkGrey"
    | "dkGreen"
    | "dkKhaki"
    | "dkMagenta"
    | "dkOliveGreen"
    | "dkOrange"
    | "dkOrchid"
    | "dkRed"
    | "dkSalmon"
    | "dkSeaGreen"
    | "dkSlateBlue"
    | "dkSlateGray"
    | "dkSlateGrey"
    | "dkTurquoise"
    | "dkViolet"
    | "deepPink"
    | "deepSkyBlue"
    | "dimGray"
    | "dimGrey"
    | "dodgerBlue"
    | "firebrick"
    | "floralWhite"
    | "forestGreen"
    | "fuchsia"
    | "gainsboro"
    | "ghostWhite"
    | "gold"
    | "goldenrod"
    | "gray"
    | "grey"
    | "green"
    | "greenYellow"
    | "honeydew"
    | "hotPink"
    | "indianRed"
    | "indigo"
    | "ivory"
    | "khaki"
    | "lavender"
    | "lavenderBlush"
    | "lawnGreen"
    | "lemonChiffon"
```



```
    | "lightBlue"
    | "lightCoral"
    | "lightCyan"
    | "lightGoldenrodYellow"
    | "lightGray"
    | "lightGrey"
    | "lightGreen"
    | "lightPink"
    | "lightSalmon"
    | "lightSeaGreen"
    | "lightSkyBlue"
    | "lightSlateGray"
    | "lightSlateGrey"
    | "lightSteelBlue"
    | "lightYellow"
    | "ltBlue"
    | "ltCoral"
    | "ltCyan"
    | "ltGoldenrodYellow"
    | "ltGray"
    | "ltGrey"
    | "ltGreen"
    | "ltPink"
    | "ltSalmon"
    | "ltSeaGreen"
    | "ltSkyBlue"
    | "ltSlateGray"
    | "ltSlateGrey"
    | "ltSteelBlue"
    | "ltYellow"
    | "lime"
    | "limeGreen"
    | "linen"
    | "magenta"
    | "maroon"
    | "medAquamarine"
    | "medBlue"
    | "medOrchid"
    | "medPurple"
    | "medSeaGreen"
    | "medSlateBlue"
    | "medSpringGreen"
    | "medTurquoise"
    | "medVioletRed"
    | "mediumAquamarine"
    | "mediumBlue"
    | "mediumOrchid"
    | "mediumPurple"
    | "mediumSeaGreen"
    | "mediumSlateBlue"
    | "mediumSpringGreen"
    | "mediumTurquoise"
    | "mediumVioletRed"
```

```
  | "midnightBlue"
  | "mintCream"
  | "mistyRose"
  | "moccasin"
  | "navajoWhite"
  | "navy"
  | "oldLace"
  | "olive"
  | "oliveDrab"
  | "orange"
  | "orangeRed"
  | "orchid"
  | "paleGoldenrod"
  | "paleGreen"
  | "paleTurquoise"
  | "paleVioletRed"
  | "papayaWhip"
  | "peachPuff"
  | "peru"
  | "pink"
  | "plum"
  | "powderBlue"
  | "purple"
  | "red"
  | "rosyBrown"
  | "royalBlue"
  | "saddleBrown"
  | "salmon"
  | "sandyBrown"
  | "seaGreen"
  | "seaShell"
  | "sienna"
  | "silver"
  | "skyBlue"
  | "slateBlue"
  | "slateGray"
  | "slateGrey"
  | "snow"
  | "springGreen"
  | "steelBlue"
  | "tan"
  | "teal"
  | "thistle"
  | "tomato"
  | "turquoise"
  | "violet"
  | "wheat"
  | "white"
  | "whiteSmoke"
  | "yellow"
  | "yellowGreen"
a_CT_PresetColor =
  attribute val { a_ST_PresetColorVal },
```



```
  a_EG_ColorTransform*
a_EG_OfficeArtExtensionList = element ext { a_CT_OfficeArtExtension }*
a_CT_OfficeArtExtensionList = a_EG_OfficeArtExtensionList
a_CT_Scale2D =
  element sx { a_CT_Ratio },
  element sy { a_CT_Ratio }
a_CT_Transform2D =

  ## default value: 0
  attribute rot { a_ST_Angle }?,

  ## default value: false
  attribute flipH { xsd:boolean }?,

  ## default value: false
  attribute flipV { xsd:boolean }?,
  element off { a_CT_Point2D }?,
  element ext { a_CT_PositiveSize2D }?
a_CT_GroupTransform2D =

  ## default value: 0
  attribute rot { a_ST_Angle }?,

  ## default value: false
  attribute flipH { xsd:boolean }?,

  ## default value: false
  attribute flipV { xsd:boolean }?,
  element off { a_CT_Point2D }?,
  element ext { a_CT_PositiveSize2D }?,
  element chOff { a_CT_Point2D }?,
  element chExt { a_CT_PositiveSize2D }?
a_CT_Point3D =
  attribute x { a_ST_Coordinate },
  attribute y { a_ST_Coordinate },
  attribute z { a_ST_Coordinate }
a_CT_Vector3D =
  attribute dx { a_ST_Coordinate },
  attribute dy { a_ST_Coordinate },
  attribute dz { a_ST_Coordinate }
a_CT_SphereCoords =
  attribute lat { a_ST_PositiveFixedAngle },
  attribute lon { a_ST_PositiveFixedAngle },
  attribute rev { a_ST_PositiveFixedAngle }
a_CT_RelativeRect =

  ## default value: 0%
  attribute l { a_ST_Percentage }?,

  ## default value: 0%
  attribute t { a_ST_Percentage }?,

  ## default value: 0%
```

```
  attribute r { a_ST_Percentage }?,

  ## default value: 0%
  attribute b { a_ST_Percentage }?
a_ST_RectAlignment =
  "tl" | "t" | "tr" | "l" | "ctr" | "r" | "bl" | "b" | "br"
a_EG_ColorChoice =
  element scrgbClr { a_CT_ScRgbColor }
  | element srgbClr { a_CT_SRgbColor }
  | element hslClr { a_CT_HslColor }
  | element sysClr { a_CT_SystemColor }
  | element schemeClr { a_CT_SchemeColor }
  | element prstClr { a_CT_PresetColor }
a_CT_Color = a_EG_ColorChoice
a_CT_ColorMRU = a_EG_ColorChoice*
a_ST_BlackWhiteMode =
  "clr"
  | "auto"
  | "gray"
  | "ltGray"
  | "invGray"
  | "grayWhite"
  | "blackGray"
  | "blackWhite"
  | "black"
  | "white"
  | "hidden"
a_AG_Blob = r_embed?, r_link?
a_CT_EmbeddedWAVAudioFile =
  r_embed,
  attribute name { xsd:string }?
a_CT_Hyperlink =
  r_id?,
  attribute invalidUrl { xsd:string }?,
  attribute action { xsd:string }?,
  attribute tgtFrame { xsd:string }?,
  attribute tooltip { xsd:string }?,

  ## default value: true
  attribute history { xsd:boolean }?,

  ## default value: false
  attribute highlightClick { xsd:boolean }?,

  ## default value: false
  attribute endSnd { xsd:boolean }?,
  element snd { a_CT_EmbeddedWAVAudioFile }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_ST_DrawingElementId = xsd:unsignedInt
a_AG_Locking =

  ## default value: false
  attribute noGrp { xsd:boolean }?,
```



```
  ## default value: false
  attribute noSelect { xsd:boolean }?,

  ## default value: false
  attribute noRot { xsd:boolean }?,

  ## default value: false
  attribute noChangeAspect { xsd:boolean }?,

  ## default value: false
  attribute noMove { xsd:boolean }?,

  ## default value: false
  attribute noResize { xsd:boolean }?,

  ## default value: false
  attribute noEditPoints { xsd:boolean }?,

  ## default value: false
  attribute noAdjustHandles { xsd:boolean }?,

  ## default value: false
  attribute noChangeArrowheads { xsd:boolean }?,

  ## default value: false
  attribute noChangeShapeType { xsd:boolean }?
a_CT_ConnectorLocking =
  a_AG_Locking,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_ShapeLocking =
  a_AG_Locking,

  ## default value: false
  attribute noTextEdit { xsd:boolean }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_PictureLocking =
  a_AG_Locking,

  ## default value: false
  attribute noCrop { xsd:boolean }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_GroupLocking =

  ## default value: false
  attribute noGrp { xsd:boolean }?,

  ## default value: false
  attribute noUngrp { xsd:boolean }?,

  ## default value: false
  attribute noSelect { xsd:boolean }?,

```

```
  ## default value: false
  attribute noRot { xsd:boolean }?,

  ## default value: false
  attribute noChangeAspect { xsd:boolean }?,

  ## default value: false
  attribute noMove { xsd:boolean }?,

  ## default value: false
  attribute noResize { xsd:boolean }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_GraphicalObjectFrameLocking =

  ## default value: false
  attribute noGrp { xsd:boolean }?,

  ## default value: false
  attribute noDrilldown { xsd:boolean }?,

  ## default value: false
  attribute noSelect { xsd:boolean }?,

  ## default value: false
  attribute noChangeAspect { xsd:boolean }?,

  ## default value: false
  attribute noMove { xsd:boolean }?,

  ## default value: false
  attribute noResize { xsd:boolean }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_ContentPartLocking =
   a_AG_Locking,
   element extLst { a_CT_OfficeArtExtensionList }?
a_CT_NonVisualDrawingProps =
  attribute id { a_ST_DrawingElementId },
  attribute name { xsd:string },
  attribute descr { xsd:string }?,

  ## default value: false
  attribute hidden { xsd:boolean }?,
  attribute title { xsd:string }?,
  element hlinkClick { a_CT_Hyperlink }?,
  element hlinkHover { a_CT_Hyperlink }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_NonVisualDrawingShapeProps =

  ## default value: false
  attribute txBox { xsd:boolean }?,
  element spLocks { a_CT_ShapeLocking }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_NonVisualConnectorProperties =
```



```
  element cxnSpLocks { a_CT_ConnectorLocking }?,
  element stCxn { a_CT_Connection }?,
  element endCxn { a_CT_Connection }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_NonVisualPictureProperties =

  ## default value: true
  attribute preferRelativeResize { xsd:boolean }?,
  element picLocks { a_CT_PictureLocking }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_NonVisualGroupDrawingShapeProps =
  element grpSpLocks { a_CT_GroupLocking }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_NonVisualGraphicFrameProperties =
  element graphicFrameLocks { a_CT_GraphicalObjectFrameLocking }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_NonVisualContentPartProperties =

   ## default value: true
   attribute isComment { xsd:boolean }?,
   element cpLocks { a_CT_ContentPartLocking }?,
   element extLst { a_CT_OfficeArtExtensionList }?
a_CT_GraphicalObjectData =
  attribute uri { xsd:token },
  a_CT_GraphicalObjectData_any*
a_CT_GraphicalObjectData_any =
  element * - (o:* | v:* | w10:* | x:*) {
    anyAttribute*,
    mixed { anyElement* }
  }
a_CT_GraphicalObject = element graphicData { a_CT_GraphicalObjectData }
a_graphic = element graphic { a_CT_GraphicalObject }
a_ST_ChartBuildStep =
  "category"
  | "ptInCategory"
  | "series"
  | "ptInSeries"
  | "allPts"
  | "gridLegend"
a_ST_DgmBuildStep = "sp" | "bg"
a_CT_AnimationDgmElement =

  ## default value: {00000000-0000-0000-0000-000000000000}
  attribute id { s_ST_Guid }?,

  ## default value: sp
  attribute bldStep { a_ST_DgmBuildStep }?
a_CT_AnimationChartElement =

  ## default value: -1
  attribute seriesIdx { xsd:int }?,

  ## default value: -1
```

```
  attribute categoryIdx { xsd:int }?,
  attribute bldStep { a_ST_ChartBuildStep }
a_CT_AnimationElementChoice =
  element dgm { a_CT_AnimationDgmElement }
  | element chart { a_CT_AnimationChartElement }
a_ST_AnimationBuildType = "allAtOnce"
a_ST_AnimationDgmOnlyBuildType = "one" | "lvlOne" | "lvlAtOnce"
a_ST_AnimationDgmBuildType =
  a_ST_AnimationBuildType | a_ST_AnimationDgmOnlyBuildType
a_CT_AnimationDgmBuildProperties =

  ## default value: allAtOnce
  attribute bld { a_ST_AnimationDgmBuildType }?,

  ## default value: false
  attribute rev { xsd:boolean }?
a_ST_AnimationChartOnlyBuildType =
  "series" | "category" | "seriesEl" | "categoryEl"
a_ST_AnimationChartBuildType =
  a_ST_AnimationBuildType | a_ST_AnimationChartOnlyBuildType
a_CT_AnimationChartBuildProperties =

  ## default value: allAtOnce
  attribute bld { a_ST_AnimationChartBuildType }?,

  ## default value: true
  attribute animBg { xsd:boolean }?
a_CT_AnimationGraphicalObjectBuildProperties =
  element bldDgm { a_CT_AnimationDgmBuildProperties }
  | element bldChart { a_CT_AnimationChartBuildProperties }
a_CT_BackgroundFormatting = a_EG_FillProperties?, a_EG_EffectProperties?
a_CT_WholeE2oFormatting =
  element ln { a_CT_LineProperties }?,
  a_EG_EffectProperties?
a_CT_GvmlUseShapeRectangle = empty
a_CT_GvmlTextShape =
  element txBody { a_CT_TextBody },
  (element useSpRect { a_CT_GvmlUseShapeRectangle }
   | element xfrm { a_CT_Transform2D }),
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_GvmlShapeNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvSpPr { a_CT_NonVisualDrawingShapeProps }
a_CT_GvmlShape =
  element nvSpPr { a_CT_GvmlShapeNonVisual },
  element spPr { a_CT_ShapeProperties },
  element txSp { a_CT_GvmlTextShape }?,
  element style { a_CT_ShapeStyle }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_GvmlConnectorNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvCxnSpPr { a_CT_NonVisualConnectorProperties }
a_CT_GvmlConnector =
```



```
  element nvCxnSpPr { a_CT_GvmlConnectorNonVisual },
  element spPr { a_CT_ShapeProperties },
  element style { a_CT_ShapeStyle }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_GvmlPictureNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvPicPr { a_CT_NonVisualPictureProperties }
a_CT_GvmlPicture =
  element nvPicPr { a_CT_GvmlPictureNonVisual },
  element blipFill { a_CT_BlipFillProperties },
  element spPr { a_CT_ShapeProperties },
  element style { a_CT_ShapeStyle }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_GvmlGraphicFrameNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvGraphicFramePr { a_CT_NonVisualGraphicFrameProperties }
a_CT_GvmlGraphicalObjectFrame =
  element nvGraphicFramePr { a_CT_GvmlGraphicFrameNonVisual },
  a_graphic,
  element xfrm { a_CT_Transform2D },
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_GvmlGroupShapeNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvGrpSpPr { a_CT_NonVisualGroupDrawingShapeProps }
a_CT_GvmlGroupShape =
  element nvGrpSpPr { a_CT_GvmlGroupShapeNonVisual },
  element grpSpPr { a_CT_GroupShapeProperties },
  (element txSp { a_CT_GvmlTextShape }
   | element sp { a_CT_GvmlShape }
   | element cxnSp { a_CT_GvmlConnector }
   | element pic { a_CT_GvmlPicture }
   | element graphicFrame { a_CT_GvmlGraphicalObjectFrame }
   | element grpSp { a_CT_GvmlGroupShape })*,
  element extLst { a_CT_OfficeArtExtensionList }?
a_ST_PresetCameraType =
  "legacyObliqueTopLeft"
  | "legacyObliqueTop"
  | "legacyObliqueTopRight"
  | "legacyObliqueLeft"
  | "legacyObliqueFront"
  | "legacyObliqueRight"
  | "legacyObliqueBottomLeft"
  | "legacyObliqueBottom"
  | "legacyObliqueBottomRight"
  | "legacyPerspectiveTopLeft"
  | "legacyPerspectiveTop"
  | "legacyPerspectiveTopRight"
  | "legacyPerspectiveLeft"
  | "legacyPerspectiveFront"
  | "legacyPerspectiveRight"
  | "legacyPerspectiveBottomLeft"
  | "legacyPerspectiveBottom"
  | "legacyPerspectiveBottomRight"
```

```
  | "orthographicFront"
  | "isometricTopUp"
  | "isometricTopDown"
  | "isometricBottomUp"
  | "isometricBottomDown"
  | "isometricLeftUp"
  | "isometricLeftDown"
  | "isometricRightUp"
  | "isometricRightDown"
  | "isometricOffAxis1Left"
  | "isometricOffAxis1Right"
  | "isometricOffAxis1Top"
  | "isometricOffAxis2Left"
  | "isometricOffAxis2Right"
  | "isometricOffAxis2Top"
  | "isometricOffAxis3Left"
  | "isometricOffAxis3Right"
  | "isometricOffAxis3Bottom"
  | "isometricOffAxis4Left"
  | "isometricOffAxis4Right"
  | "isometricOffAxis4Bottom"
  | "obliqueTopLeft"
  | "obliqueTop"
  | "obliqueTopRight"
  | "obliqueLeft"
  | "obliqueRight"
  | "obliqueBottomLeft"
  | "obliqueBottom"
  | "obliqueBottomRight"
  | "perspectiveFront"
  | "perspectiveLeft"
  | "perspectiveRight"
  | "perspectiveAbove"
  | "perspectiveBelow"
  | "perspectiveAboveLeftFacing"
  | "perspectiveAboveRightFacing"
  | "perspectiveContrastingLeftFacing"
  | "perspectiveContrastingRightFacing"
  | "perspectiveHeroicLeftFacing"
  | "perspectiveHeroicRightFacing"
  | "perspectiveHeroicExtremeLeftFacing"
  | "perspectiveHeroicExtremeRightFacing"
  | "perspectiveRelaxed"
  | "perspectiveRelaxedModerately"
a_ST_FOVAngle = xsd:int { minInclusive = "0" maxInclusive = "10800000" }
a_CT_Camera =
  attribute prst { a_ST_PresetCameraType },
  attribute fov { a_ST_FOVAngle }?,

  ## default value: 100%
  attribute zoom { a_ST_PositivePercentage }?,
  element rot { a_CT_SphereCoords }?
a_ST_LightRigDirection =
```



```
  "tl" | "t" | "tr" | "l" | "r" | "bl" | "b" | "br"
a_ST_LightRigType =
  "legacyFlat1"
  | "legacyFlat2"
  | "legacyFlat3"
  | "legacyFlat4"
  | "legacyNormal1"
  | "legacyNormal2"
  | "legacyNormal3"
  | "legacyNormal4"
  | "legacyHarsh1"
  | "legacyHarsh2"
  | "legacyHarsh3"
  | "legacyHarsh4"
  | "threePt"
  | "balanced"
  | "soft"
  | "harsh"
  | "flood"
  | "contrasting"
  | "morning"
  | "sunrise"
  | "sunset"
  | "chilly"
  | "freezing"
  | "flat"
  | "twoPt"
  | "glow"
  | "brightRoom"
a_CT_LightRig =
  attribute rig { a_ST_LightRigType },
  attribute dir { a_ST_LightRigDirection },
  element rot { a_CT_SphereCoords }?
a_CT_Scene3D =
  element camera { a_CT_Camera },
  element lightRig { a_CT_LightRig },
  element backdrop { a_CT_Backdrop }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_Backdrop =
  element anchor { a_CT_Point3D },
  element norm { a_CT_Vector3D },
  element up { a_CT_Vector3D },
  element extLst { a_CT_OfficeArtExtensionList }?
a_ST_BevelPresetType =
  "relaxedInset"
  | "circle"
  | "slope"
  | "cross"
  | "angle"
  | "softRound"
  | "convex"
  | "coolSlant"
  | "divot"
```

```
  | "riblet"
  | "hardEdge"
  | "artDeco"
a_CT_Bevel =

  ## default value: 76200
  attribute w { a_ST_PositiveCoordinate }?,

  ## default value: 76200
  attribute h { a_ST_PositiveCoordinate }?,

  ## default value: circle
  attribute prst { a_ST_BevelPresetType }?
a_ST_PresetMaterialType =
  "legacyMatte"
  | "legacyPlastic"
  | "legacyMetal"
  | "legacyWireframe"
  | "matte"
  | "plastic"
  | "metal"
  | "warmMatte"
  | "translucentPowder"
  | "powder"
  | "dkEdge"
  | "softEdge"
  | "clear"
  | "flat"
  | "softmetal"
a_CT_Shape3D =

  ## default value: 0
  attribute z { a_ST_Coordinate }?,

  ## default value: 0
  attribute extrusionH { a_ST_PositiveCoordinate }?,

  ## default value: 0
  attribute contourW { a_ST_PositiveCoordinate }?,

  ## default value: warmMatte
  attribute prstMaterial { a_ST_PresetMaterialType }?,
  element bevelT { a_CT_Bevel }?,
  element bevelB { a_CT_Bevel }?,
  element extrusionClr { a_CT_Color }?,
  element contourClr { a_CT_Color }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_FlatText =

  ## default value: 0
  attribute z { a_ST_Coordinate }?
a_EG_Text3D =
  element sp3d { a_CT_Shape3D }
```



```
  | element flatTx { a_CT_FlatText }
a_CT_AlphaBiLevelEffect =
  attribute thresh { a_ST_PositiveFixedPercentage }
a_CT_AlphaCeilingEffect = empty
a_CT_AlphaFloorEffect = empty
a_CT_AlphaInverseEffect = a_EG_ColorChoice?
a_CT_AlphaModulateFixedEffect =

  ## default value: 100%
  attribute amt { a_ST_PositivePercentage }?
a_CT_AlphaOutsetEffect =

  ## default value: 0
  attribute rad { a_ST_Coordinate }?
a_CT_AlphaReplaceEffect = attribute a { a_ST_PositiveFixedPercentage }
a_CT_BiLevelEffect = attribute thresh { a_ST_PositiveFixedPercentage }
a_CT_BlurEffect =

  ## default value: 0
  attribute rad { a_ST_PositiveCoordinate }?,

  ## default value: true
  attribute grow { xsd:boolean }?
a_CT_ColorChangeEffect =

  ## default value: true
  attribute useA { xsd:boolean }?,
  element clrFrom { a_CT_Color },
  element clrTo { a_CT_Color }
a_CT_ColorReplaceEffect = a_EG_ColorChoice
a_CT_DuotoneEffect = a_EG_ColorChoice+
a_CT_GlowEffect =

  ## default value: 0
  attribute rad { a_ST_PositiveCoordinate }?,
  a_EG_ColorChoice
a_CT_GrayscaleEffect = empty
a_CT_HSLEffect =

  ## default value: 0
  attribute hue { a_ST_PositiveFixedAngle }?,

  ## default value: 0%
  attribute sat { a_ST_FixedPercentage }?,

  ## default value: 0%
  attribute lum { a_ST_FixedPercentage }?
a_CT_InnerShadowEffect =

  ## default value: 0
  attribute blurRad { a_ST_PositiveCoordinate }?,

  ## default value: 0
```

```
  attribute dist { a_ST_PositiveCoordinate }?,

  ## default value: 0
  attribute dir { a_ST_PositiveFixedAngle }?,
  a_EG_ColorChoice
a_CT_LuminanceEffect =

  ## default value: 0%
  attribute bright { a_ST_FixedPercentage }?,

  ## default value: 0%
  attribute contrast { a_ST_FixedPercentage }?
a_CT_OuterShadowEffect =

  ## default value: 0
  attribute blurRad { a_ST_PositiveCoordinate }?,

  ## default value: 0
  attribute dist { a_ST_PositiveCoordinate }?,

  ## default value: 0
  attribute dir { a_ST_PositiveFixedAngle }?,

  ## default value: 100%
  attribute sx { a_ST_Percentage }?,

  ## default value: 100%
  attribute sy { a_ST_Percentage }?,

  ## default value: 0
  attribute kx { a_ST_FixedAngle }?,

  ## default value: 0
  attribute ky { a_ST_FixedAngle }?,

  ## default value: b
  attribute algn { a_ST_RectAlignment }?,

  ## default value: true
  attribute rotWithShape { xsd:boolean }?,
  a_EG_ColorChoice
a_ST_PresetShadowVal =
  "shdw1"
  | "shdw2"
  | "shdw3"
  | "shdw4"
  | "shdw5"
  | "shdw6"
  | "shdw7"
  | "shdw8"
  | "shdw9"
  | "shdw10"
  | "shdw11"
```



```
  | "shdw12"
  | "shdw13"
  | "shdw14"
  | "shdw15"
  | "shdw16"
  | "shdw17"
  | "shdw18"
  | "shdw19"
  | "shdw20"
a_CT_PresetShadowEffect =
  attribute prst { a_ST_PresetShadowVal },

  ## default value: 0
  attribute dist { a_ST_PositiveCoordinate }?,

  ## default value: 0
  attribute dir { a_ST_PositiveFixedAngle }?,
  a_EG_ColorChoice
a_CT_ReflectionEffect =

  ## default value: 0
  attribute blurRad { a_ST_PositiveCoordinate }?,

  ## default value: 100%
  attribute stA { a_ST_PositiveFixedPercentage }?,

  ## default value: 0%
  attribute stPos { a_ST_PositiveFixedPercentage }?,

  ## default value: 0%
  attribute endA { a_ST_PositiveFixedPercentage }?,

  ## default value: 100%
  attribute endPos { a_ST_PositiveFixedPercentage }?,

  ## default value: 0
  attribute dist { a_ST_PositiveCoordinate }?,

  ## default value: 0
  attribute dir { a_ST_PositiveFixedAngle }?,

  ## default value: 5400000
  attribute fadeDir { a_ST_PositiveFixedAngle }?,

  ## default value: 100%
  attribute sx { a_ST_Percentage }?,

  ## default value: 100%
  attribute sy { a_ST_Percentage }?,

  ## default value: 0
  attribute kx { a_ST_FixedAngle }?,

```

```
  ## default value: 0
  attribute ky { a_ST_FixedAngle }?,

  ## default value: b
  attribute algn { a_ST_RectAlignment }?,

  ## default value: true
  attribute rotWithShape { xsd:boolean }?
a_CT_RelativeOffsetEffect =

  ## default value: 0%
  attribute tx { a_ST_Percentage }?,

  ## default value: 0%
  attribute ty { a_ST_Percentage }?
a_CT_SoftEdgesEffect = attribute rad { a_ST_PositiveCoordinate }
a_CT_TintEffect =

  ## default value: 0
  attribute hue { a_ST_PositiveFixedAngle }?,

  ## default value: 0%
  attribute amt { a_ST_FixedPercentage }?
a_CT_TransformEffect =

  ## default value: 100%
  attribute sx { a_ST_Percentage }?,

  ## default value: 100%
  attribute sy { a_ST_Percentage }?,

  ## default value: 0
  attribute kx { a_ST_FixedAngle }?,

  ## default value: 0
  attribute ky { a_ST_FixedAngle }?,

  ## default value: 0
  attribute tx { a_ST_Coordinate }?,

  ## default value: 0
  attribute ty { a_ST_Coordinate }?
a_CT_NoFillProperties = empty
a_CT_SolidColorFillProperties = a_EG_ColorChoice?
a_CT_LinearShadeProperties =
  attribute ang { a_ST_PositiveFixedAngle }?,
  attribute scaled { xsd:boolean }?
a_ST_PathShadeType = "shape" | "circle" | "rect"
a_CT_PathShadeProperties =
  attribute path { a_ST_PathShadeType }?,
  element fillToRect { a_CT_RelativeRect }?
a_EG_ShadeProperties =
  element lin { a_CT_LinearShadeProperties }
```



```
  | element path { a_CT_PathShadeProperties }
a_ST_TileFlipMode = "none" | "x" | "y" | "xy"
a_CT_GradientStop =
  attribute pos { a_ST_PositiveFixedPercentage },
  a_EG_ColorChoice
a_CT_GradientStopList = element gs { a_CT_GradientStop }+
a_CT_GradientFillProperties =
  attribute flip { a_ST_TileFlipMode }?,
  attribute rotWithShape { xsd:boolean }?,
  element gsLst { a_CT_GradientStopList }?,
  a_EG_ShadeProperties?,
  element tileRect { a_CT_RelativeRect }?
a_CT_TileInfoProperties =
  attribute tx { a_ST_Coordinate }?,
  attribute ty { a_ST_Coordinate }?,
  attribute sx { a_ST_Percentage }?,
  attribute sy { a_ST_Percentage }?,
  attribute flip { a_ST_TileFlipMode }?,
  attribute algn { a_ST_RectAlignment }?
a_CT_StretchInfoProperties = element fillRect { a_CT_RelativeRect }?
a_EG_FillModeProperties =
  element tile { a_CT_TileInfoProperties }
  | element stretch { a_CT_StretchInfoProperties }
a_ST_BlipCompression = "email" | "screen" | "print" | "hqprint" | "none"
a_CT_Blip =
  a_AG_Blob,

  ## default value: none
  attribute cstate { a_ST_BlipCompression }?,
  (element alphaBiLevel { a_CT_AlphaBiLevelEffect }
   | element alphaCeiling { a_CT_AlphaCeilingEffect }
   | element alphaFloor { a_CT_AlphaFloorEffect }
   | element alphaInv { a_CT_AlphaInverseEffect }
   | element alphaMod { a_CT_AlphaModulateEffect }
   | element alphaModFix { a_CT_AlphaModulateFixedEffect }
   | element alphaRepl { a_CT_AlphaReplaceEffect }
   | element biLevel { a_CT_BiLevelEffect }
   | element blur { a_CT_BlurEffect }
   | element clrChange { a_CT_ColorChangeEffect }
   | element clrRepl { a_CT_ColorReplaceEffect }
   | element duotone { a_CT_DuotoneEffect }
   | element fillOverlay { a_CT_FillOverlayEffect }
   | element grayscl { a_CT_GrayscaleEffect }
   | element hsl { a_CT_HSLEffect }
   | element lum { a_CT_LuminanceEffect }
   | element tint { a_CT_TintEffect })*,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_BlipFillProperties =
  attribute dpi { xsd:unsignedInt }?,
  attribute rotWithShape { xsd:boolean }?,
  element blip { a_CT_Blip }?,
  element srcRect { a_CT_RelativeRect }?,
  a_EG_FillModeProperties?
```

```
a_ST_PresetPatternVal =
  "pct5"
  | "pct10"
  | "pct20"
  | "pct25"
  | "pct30"
  | "pct40"
  | "pct50"
  | "pct60"
  | "pct70"
  | "pct75"
  | "pct80"
  | "pct90"
  | "horz"
  | "vert"
  | "ltHorz"
  | "ltVert"
  | "dkHorz"
  | "dkVert"
  | "narHorz"
  | "narVert"
  | "dashHorz"
  | "dashVert"
  | "cross"
  | "dnDiag"
  | "upDiag"
  | "ltDnDiag"
  | "ltUpDiag"
  | "dkDnDiag"
  | "dkUpDiag"
  | "wdDnDiag"
  | "wdUpDiag"
  | "dashDnDiag"
  | "dashUpDiag"
  | "diagCross"
  | "smCheck"
  | "lgCheck"
  | "smGrid"
  | "lgGrid"
  | "dotGrid"
  | "smConfetti"
  | "lgConfetti"
  | "horzBrick"
  | "diagBrick"
  | "solidDmnd"
  | "openDmnd"
  | "dotDmnd"
  | "plaid"
  | "sphere"
  | "weave"
  | "divot"
  | "shingle"
  | "wave"
```



```
  | "trellis"
  | "zigZag"
a_CT_PatternFillProperties =
  attribute prst { a_ST_PresetPatternVal }?,
  element fgClr { a_CT_Color }?,
  element bgClr { a_CT_Color }?
a_CT_GroupFillProperties = empty
a_EG_FillProperties =
  element noFill { a_CT_NoFillProperties }
  | element solidFill { a_CT_SolidColorFillProperties }
  | element gradFill { a_CT_GradientFillProperties }
  | element blipFill { a_CT_BlipFillProperties }
  | element pattFill { a_CT_PatternFillProperties }
  | element grpFill { a_CT_GroupFillProperties }
a_CT_FillProperties = a_EG_FillProperties
a_CT_FillEffect = a_EG_FillProperties
a_ST_BlendMode = "over" | "mult" | "screen" | "darken" | "lighten"
a_CT_FillOverlayEffect =
  attribute blend { a_ST_BlendMode },
  a_EG_FillProperties
a_CT_EffectReference = attribute ref { xsd:token }
a_EG_Effect =
  element cont { a_CT_EffectContainer }
  | element effect { a_CT_EffectReference }
  | element alphaBiLevel { a_CT_AlphaBiLevelEffect }
  | element alphaCeiling { a_CT_AlphaCeilingEffect }
  | element alphaFloor { a_CT_AlphaFloorEffect }
  | element alphaInv { a_CT_AlphaInverseEffect }
  | element alphaMod { a_CT_AlphaModulateEffect }
  | element alphaModFix { a_CT_AlphaModulateFixedEffect }
  | element alphaOutset { a_CT_AlphaOutsetEffect }
  | element alphaRepl { a_CT_AlphaReplaceEffect }
  | element biLevel { a_CT_BiLevelEffect }
  | element blend { a_CT_BlendEffect }
  | element blur { a_CT_BlurEffect }
  | element clrChange { a_CT_ColorChangeEffect }
  | element clrRepl { a_CT_ColorReplaceEffect }
  | element duotone { a_CT_DuotoneEffect }
  | element fill { a_CT_FillEffect }
  | element fillOverlay { a_CT_FillOverlayEffect }
  | element glow { a_CT_GlowEffect }
  | element grayscl { a_CT_GrayscaleEffect }
  | element hsl { a_CT_HSLEffect }
  | element innerShdw { a_CT_InnerShadowEffect }
  | element lum { a_CT_LuminanceEffect }
  | element outerShdw { a_CT_OuterShadowEffect }
  | element prstShdw { a_CT_PresetShadowEffect }
  | element reflection { a_CT_ReflectionEffect }
  | element relOff { a_CT_RelativeOffsetEffect }
  | element softEdge { a_CT_SoftEdgesEffect }
  | element tint { a_CT_TintEffect }
  | element xfrm { a_CT_TransformEffect }
a_ST_EffectContainerType = "sib" | "tree"
```

```
a_CT_EffectContainer =

  ## default value: sib
  attribute type { a_ST_EffectContainerType }?,
  attribute name { xsd:token }?,
  a_EG_Effect*
a_CT_AlphaModulateEffect = element cont { a_CT_EffectContainer }
a_CT_BlendEffect =
  attribute blend { a_ST_BlendMode },
  element cont { a_CT_EffectContainer }
a_CT_EffectList =
  element blur { a_CT_BlurEffect }?,
  element fillOverlay { a_CT_FillOverlayEffect }?,
  element glow { a_CT_GlowEffect }?,
  element innerShdw { a_CT_InnerShadowEffect }?,
  element outerShdw { a_CT_OuterShadowEffect }?,
  element prstShdw { a_CT_PresetShadowEffect }?,
  element reflection { a_CT_ReflectionEffect }?,
  element softEdge { a_CT_SoftEdgesEffect }?
a_EG_EffectProperties =
  element effectLst { a_CT_EffectList }
  | element effectDag { a_CT_EffectContainer }
a_CT_EffectProperties = a_EG_EffectProperties
a_blip = element blip { a_CT_Blip }
a_ST_ShapeType =
  "line"
  | "lineInv"
  | "triangle"
  | "rtTriangle"
  | "rect"
  | "diamond"
  | "parallelogram"
  | "trapezoid"
  | "nonIsoscelesTrapezoid"
  | "pentagon"
  | "hexagon"
  | "heptagon"
  | "octagon"
  | "decagon"
  | "dodecagon"
  | "star4"
  | "star5"
  | "star6"
  | "star7"
  | "star8"
  | "star10"
  | "star12"
  | "star16"
  | "star24"
  | "star32"
  | "roundRect"
  | "round1Rect"
  | "round2SameRect"
```



```
    | "round2DiagRect"
    | "snipRoundRect"
    | "snip1Rect"
    | "snip2SameRect"
    | "snip2DiagRect"
    | "plaque"
    | "ellipse"
    | "teardrop"
    | "homePlate"
    | "chevron"
    | "pieWedge"
    | "pie"
    | "blockArc"
    | "donut"
    | "noSmoking"
    | "rightArrow"
    | "leftArrow"
    | "upArrow"
    | "downArrow"
    | "stripedRightArrow"
    | "notchedRightArrow"
    | "bentUpArrow"
    | "leftRightArrow"
    | "upDownArrow"
    | "leftUpArrow"
    | "leftRightUpArrow"
    | "quadArrow"
    | "leftArrowCallout"
    | "rightArrowCallout"
    | "upArrowCallout"
    | "downArrowCallout"
    | "leftRightArrowCallout"
    | "upDownArrowCallout"
    | "quadArrowCallout"
    | "bentArrow"
    | "uturnArrow"
    | "circularArrow"
    | "leftCircularArrow"
    | "leftRightCircularArrow"
    | "curvedRightArrow"
    | "curvedLeftArrow"
    | "curvedUpArrow"
    | "curvedDownArrow"
    | "swooshArrow"
    | "cube"
    | "can"
    | "lightningBolt"
    | "heart"
    | "sun"
    | "moon"
    | "smileyFace"
    | "irregularSeal1"
    | "irregularSeal2"
```

```
    | "foldedCorner"
    | "bevel"
    | "frame"
    | "halfFrame"
    | "corner"
    | "diagStripe"
    | "chord"
    | "arc"
    | "leftBracket"
    | "rightBracket"
    | "leftBrace"
    | "rightBrace"
    | "bracketPair"
    | "bracePair"
    | "straightConnector1"
    | "bentConnector2"
    | "bentConnector3"
    | "bentConnector4"
    | "bentConnector5"
    | "curvedConnector2"
    | "curvedConnector3"
    | "curvedConnector4"
    | "curvedConnector5"
    | "callout1"
    | "callout2"
    | "callout3"
    | "accentCallout1"
    | "accentCallout2"
    | "accentCallout3"
    | "borderCallout1"
    | "borderCallout2"
    | "borderCallout3"
    | "accentBorderCallout1"
    | "accentBorderCallout2"
    | "accentBorderCallout3"
    | "wedgeRectCallout"
    | "wedgeRoundRectCallout"
    | "wedgeEllipseCallout"
    | "cloudCallout"
    | "cloud"
    | "ribbon"
    | "ribbon2"
    | "ellipseRibbon"
    | "ellipseRibbon2"
    | "leftRightRibbon"
    | "verticalScroll"
    | "horizontalScroll"
    | "wave"
    | "doubleWave"
    | "plus"
    | "flowChartProcess"
    | "flowChartDecision"
    | "flowChartInputOutput"
```



```
    | "flowChartPredefinedProcess"
    | "flowChartInternalStorage"
    | "flowChartDocument"
    | "flowChartMultidocument"
    | "flowChartTerminator"
    | "flowChartPreparation"
    | "flowChartManualInput"
    | "flowChartManualOperation"
    | "flowChartConnector"
    | "flowChartPunchedCard"
    | "flowChartPunchedTape"
    | "flowChartSummingJunction"
    | "flowChartOr"
    | "flowChartCollate"
    | "flowChartSort"
    | "flowChartExtract"
    | "flowChartMerge"
    | "flowChartOfflineStorage"
    | "flowChartOnlineStorage"
    | "flowChartMagneticTape"
    | "flowChartMagneticDisk"
    | "flowChartMagneticDrum"
    | "flowChartDisplay"
    | "flowChartDelay"
    | "flowChartAlternateProcess"
    | "flowChartOffpageConnector"
    | "actionButtonBlank"
    | "actionButtonHome"
    | "actionButtonHelp"
    | "actionButtonInformation"
    | "actionButtonForwardNext"
    | "actionButtonBackPrevious"
    | "actionButtonEnd"
    | "actionButtonBeginning"
    | "actionButtonReturn"
    | "actionButtonDocument"
    | "actionButtonSound"
    | "actionButtonMovie"
    | "gear6"
    | "gear9"
    | "funnel"
    | "mathPlus"
    | "mathMinus"
    | "mathMultiply"
    | "mathDivide"
    | "mathEqual"
    | "mathNotEqual"
    | "cornerTabs"
    | "squareTabs"
    | "plaqueTabs"
    | "chartX"
    | "chartStar"
    | "chartPlus"
```

```
a_ST_TextShapeType =
  "textNoShape"
  | "textPlain"
  | "textStop"
  | "textTriangle"
  | "textTriangleInverted"
  | "textChevron"
  | "textChevronInverted"
  | "textRingInside"
  | "textRingOutside"
  | "textArchUp"
  | "textArchDown"
  | "textCircle"
  | "textButton"
  | "textArchUpPour"
  | "textArchDownPour"
  | "textCirclePour"
  | "textButtonPour"
  | "textCurveUp"
  | "textCurveDown"
  | "textCanUp"
  | "textCanDown"
  | "textWave1"
  | "textWave2"
  | "textDoubleWave1"
  | "textWave4"
  | "textInflate"
  | "textDeflate"
  | "textInflateBottom"
  | "textDeflateBottom"
  | "textInflateTop"
  | "textDeflateTop"
  | "textDeflateInflate"
  | "textDeflateInflateDeflate"
  | "textFadeRight"
  | "textFadeLeft"
  | "textFadeUp"
  | "textFadeDown"
  | "textSlantUp"
  | "textSlantDown"
  | "textCascadeUp"
  | "textCascadeDown"
a_ST_GeomGuideName = xsd:token
a_ST_GeomGuideFormula = xsd:string
a_CT_GeomGuide =
  attribute name { a_ST_GeomGuideName },
  attribute fmla { a_ST_GeomGuideFormula }
a_CT_GeomGuideList = element gd { a_CT_GeomGuide }*
a_ST_AdjCoordinate = a_ST_Coordinate | a_ST_GeomGuideName
a_ST_AdjAngle = a_ST_Angle | a_ST_GeomGuideName
a_CT_AdjPoint2D =
  attribute x { a_ST_AdjCoordinate },
  attribute y { a_ST_AdjCoordinate }
```



```
a_CT_GeomRect =
  attribute l { a_ST_AdjCoordinate },
  attribute t { a_ST_AdjCoordinate },
  attribute r { a_ST_AdjCoordinate },
  attribute b { a_ST_AdjCoordinate }
a_CT_XYAdjustHandle =
  attribute gdRefX { a_ST_GeomGuideName }?,
  attribute minX { a_ST_AdjCoordinate }?,
  attribute maxX { a_ST_AdjCoordinate }?,
  attribute gdRefY { a_ST_GeomGuideName }?,
  attribute minY { a_ST_AdjCoordinate }?,
  attribute maxY { a_ST_AdjCoordinate }?,
  element pos { a_CT_AdjPoint2D }
a_CT_PolarAdjustHandle =
  attribute gdRefR { a_ST_GeomGuideName }?,
  attribute minR { a_ST_AdjCoordinate }?,
  attribute maxR { a_ST_AdjCoordinate }?,
  attribute gdRefAng { a_ST_GeomGuideName }?,
  attribute minAng { a_ST_AdjAngle }?,
  attribute maxAng { a_ST_AdjAngle }?,
  element pos { a_CT_AdjPoint2D }
a_CT_ConnectionSite =
  attribute ang { a_ST_AdjAngle },
  element pos { a_CT_AdjPoint2D }
a_CT_AdjustHandleList =
  (element ahXY { a_CT_XYAdjustHandle }
   | element ahPolar { a_CT_PolarAdjustHandle })*
a_CT_ConnectionSiteList = element cxn { a_CT_ConnectionSite }*
a_CT_Connection =
  attribute id { a_ST_DrawingElementId },
  attribute idx { xsd:unsignedInt }
a_CT_Path2DMoveTo = element pt { a_CT_AdjPoint2D }
a_CT_Path2DLineTo = element pt { a_CT_AdjPoint2D }
a_CT_Path2DArcTo =
  attribute wR { a_ST_AdjCoordinate },
  attribute hR { a_ST_AdjCoordinate },
  attribute stAng { a_ST_AdjAngle },
  attribute swAng { a_ST_AdjAngle }
a_CT_Path2DQuadBezierTo = element pt { a_CT_AdjPoint2D }+
a_CT_Path2DCubicBezierTo = element pt { a_CT_AdjPoint2D }+
a_CT_Path2DClose = empty
a_ST_PathFillMode =
  "none" | "norm" | "lighten" | "lightenLess" | "darken" | "darkenLess"
a_CT_Path2D =

  ## default value: 0
  attribute w { a_ST_PositiveCoordinate }?,

  ## default value: 0
  attribute h { a_ST_PositiveCoordinate }?,

  ## default value: norm
  attribute fill { a_ST_PathFillMode }?,
```

```

  ## default value: true
  attribute stroke { xsd:boolean }?,

  ## default value: true
  attribute extrusionOk { xsd:boolean }?,
  (element close { a_CT_Path2DClose }
   | element moveTo { a_CT_Path2DMoveTo }
   | element lnTo { a_CT_Path2DLineTo }
   | element arcTo { a_CT_Path2DArcTo }
   | element quadBezTo { a_CT_Path2DQuadBezierTo }
   | element cubicBezTo { a_CT_Path2DCubicBezierTo })*
a_CT_Path2DList = element path { a_CT_Path2D }*
a_CT_PresetGeometry2D =
  attribute prst { a_ST_ShapeType },
  element avLst { a_CT_GeomGuideList }?
a_CT_PresetTextShape =
  attribute prst { a_ST_TextShapeType },
  element avLst { a_CT_GeomGuideList }?
a_CT_CustomGeometry2D =
  element avLst { a_CT_GeomGuideList }?,
  element gdLst { a_CT_GeomGuideList }?,
  element ahLst { a_CT_AdjustHandleList }?,
  element cxnLst { a_CT_ConnectionSiteList }?,
  element rect { a_CT_GeomRect }?,
  element pathLst { a_CT_Path2DList }
a_EG_Geometry =
  element custGeom { a_CT_CustomGeometry2D }
  | element prstGeom { a_CT_PresetGeometry2D }
a_EG_TextGeometry =
  element custGeom { a_CT_CustomGeometry2D }
  | element prstTxWarp { a_CT_PresetTextShape }
a_ST_LineEndType =
  "none" | "triangle" | "stealth" | "diamond" | "oval" | "arrow"
a_ST_LineEndWidth = "sm" | "med" | "lg"
a_ST_LineEndLength = "sm" | "med" | "lg"
a_CT_LineEndProperties =
  attribute type { a_ST_LineEndType }?,
  attribute w { a_ST_LineEndWidth }?,
  attribute len { a_ST_LineEndLength }?
a_EG_LineFillProperties =
  element noFill { a_CT_NoFillProperties }
  | element solidFill { a_CT_SolidColorFillProperties }
  | element gradFill { a_CT_GradientFillProperties }
  | element pattFill { a_CT_PatternFillProperties }
a_CT_LineJoinBevel = empty
a_CT_LineJoinRound = empty
a_CT_LineJoinMiterProperties =
  attribute lim { a_ST_PositivePercentage }?
a_EG_LineJoinProperties =
  element round { a_CT_LineJoinRound }
  | element bevel { a_CT_LineJoinBevel }
  | element miter { a_CT_LineJoinMiterProperties }
```



```
a_ST_PresetLineDashVal =
  "solid"
  | "dot"
  | "dash"
  | "lgDash"
  | "dashDot"
  | "lgDashDot"
  | "lgDashDotDot"
  | "sysDash"
  | "sysDot"
  | "sysDashDot"
  | "sysDashDotDot"
a_CT_PresetLineDashProperties =
  attribute val { a_ST_PresetLineDashVal }?
a_CT_DashStop =
  attribute d { a_ST_PositivePercentage },
  attribute sp { a_ST_PositivePercentage }
a_CT_DashStopList = element ds { a_CT_DashStop }*
a_EG_LineDashProperties =
  element prstDash { a_CT_PresetLineDashProperties }
  | element custDash { a_CT_DashStopList }
a_ST_LineCap = "rnd" | "sq" | "flat"
a_ST_LineWidth =
  xsd:int { minInclusive = "0" maxInclusive = "20116800" }
a_ST_PenAlignment = "ctr" | "in"
a_ST_CompoundLine = "sng" | "dbl" | "thickThin" | "thinThick" | "tri"
a_CT_LineProperties =
  attribute w { a_ST_LineWidth }?,
  attribute cap { a_ST_LineCap }?,
  attribute cmpd { a_ST_CompoundLine }?,
  attribute algn { a_ST_PenAlignment }?,
  a_EG_LineFillProperties?,
  a_EG_LineDashProperties?,
  a_EG_LineJoinProperties?,
  element headEnd { a_CT_LineEndProperties }?,
  element tailEnd { a_CT_LineEndProperties }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_ST_ShapeID = xsd:token
a_CT_ShapeProperties =
  attribute bwMode { a_ST_BlackWhiteMode }?,
  element xfrm { a_CT_Transform2D }?,
  a_EG_Geometry?,
  a_EG_FillProperties?,
  element ln { a_CT_LineProperties }?,
  a_EG_EffectProperties?,
  element scene3d { a_CT_Scene3D }?,
  element sp3d { a_CT_Shape3D }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_GroupShapeProperties =
  attribute bwMode { a_ST_BlackWhiteMode }?,
  element xfrm { a_CT_GroupTransform2D }?,
  a_EG_FillProperties?,
  a_EG_EffectProperties?,
```

```
  element scene3d { a_CT_Scene3D }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_StyleMatrixReference =
  attribute idx { a_ST_StyleMatrixColumnIndex },
  a_EG_ColorChoice?
a_CT_FontReference =
  attribute idx { a_ST_FontCollectionIndex },
  a_EG_ColorChoice?
a_CT_ShapeStyle =
  element lnRef { a_CT_StyleMatrixReference },
  element fillRef { a_CT_StyleMatrixReference },
  element effectRef { a_CT_StyleMatrixReference },
  element fontRef { a_CT_FontReference }
a_CT_DefaultShapeDefinition =
  element spPr { a_CT_ShapeProperties },
  element bodyPr { a_CT_TextBodyProperties },
  element lstStyle { a_CT_TextListStyle },
  element style { a_CT_ShapeStyle }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_ObjectStyleDefaults =
  element spDef { a_CT_DefaultShapeDefinition }?,
  element lnDef { a_CT_DefaultShapeDefinition }?,
  element txDef { a_CT_DefaultShapeDefinition }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_EmptyElement = empty
a_CT_ColorMapping =
  attribute bg1 { a_ST_ColorSchemeIndex },
  attribute tx1 { a_ST_ColorSchemeIndex },
  attribute bg2 { a_ST_ColorSchemeIndex },
  attribute tx2 { a_ST_ColorSchemeIndex },
  attribute accent1 { a_ST_ColorSchemeIndex },
  attribute accent2 { a_ST_ColorSchemeIndex },
  attribute accent3 { a_ST_ColorSchemeIndex },
  attribute accent4 { a_ST_ColorSchemeIndex },
  attribute accent5 { a_ST_ColorSchemeIndex },
  attribute accent6 { a_ST_ColorSchemeIndex },
  attribute hlink { a_ST_ColorSchemeIndex },
  attribute folHlink { a_ST_ColorSchemeIndex },
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_ColorMappingOverride =
  element masterClrMapping { a_CT_EmptyElement }
  | element overrideClrMapping { a_CT_ColorMapping }
a_CT_ColorSchemeAndMapping =
  element clrScheme { a_CT_ColorScheme },
  element clrMap { a_CT_ColorMapping }?
a_CT_ColorSchemeList =
  element extraClrScheme { a_CT_ColorSchemeAndMapping }*
a_CT_OfficeStyleSheet =
  attribute name { xsd:string }?,
  element themeElements { a_CT_BaseStyles },
  element objectDefaults { a_CT_ObjectStyleDefaults }?,
  element extraClrSchemeLst { a_CT_ColorSchemeList }?,
  element custClrLst { a_CT_CustomColorList }?,
```



```
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_BaseStylesOverride =
  element clrScheme { a_CT_ColorScheme }?,
  element fontScheme { a_CT_FontScheme }?,
  element fmtScheme { a_CT_StyleMatrix }?
a_CT_ClipboardStyleSheet =
  element themeElements { a_CT_BaseStyles },
  element clrMap { a_CT_ColorMapping }
a_theme = element theme { a_CT_OfficeStyleSheet }
a_themeOverride = element themeOverride { a_CT_BaseStylesOverride }
a_themeManager = element themeManager { a_CT_EmptyElement }
a_CT_TableCellProperties =

  ## default value: 91440
  attribute marL { a_ST_Coordinate32 }?,

  ## default value: 91440
  attribute marR { a_ST_Coordinate32 }?,

  ## default value: 45720
  attribute marT { a_ST_Coordinate32 }?,

  ## default value: 45720
  attribute marB { a_ST_Coordinate32 }?,

  ## default value: horz
  attribute vert { a_ST_TextVerticalType }?,

  ## default value: t
  attribute anchor { a_ST_TextAnchoringType }?,

  ## default value: false
  attribute anchorCtr { xsd:boolean }?,

  ## default value: clip
  attribute horzOverflow { a_ST_TextHorzOverflowType }?,
  element lnL { a_CT_LineProperties }?,
  element lnR { a_CT_LineProperties }?,
  element lnT { a_CT_LineProperties }?,
  element lnB { a_CT_LineProperties }?,
  element lnTlToBr { a_CT_LineProperties }?,
  element lnBlToTr { a_CT_LineProperties }?,
  element cell3D { a_CT_Cell3D }?,
  a_EG_FillProperties?,
  element headers { a_CT_Headers }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_Headers = element header { xsd:string }*
a_CT_TableCol =
  attribute w { a_ST_Coordinate },
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_TableGrid = element gridCol { a_CT_TableCol }*
a_CT_TableCell =

```

```
  ## default value: 1
  attribute rowSpan { xsd:int }?,

  ## default value: 1
  attribute gridSpan { xsd:int }?,

  ## default value: false
  attribute hMerge { xsd:boolean }?,

  ## default value: false
  attribute vMerge { xsd:boolean }?,
  attribute id { xsd:string }?,
  element txBody { a_CT_TextBody }?,
  element tcPr { a_CT_TableCellProperties }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_TableRow =
  attribute h { a_ST_Coordinate },
  element tc { a_CT_TableCell }*,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_TableProperties =

  ## default value: false
  attribute rtl { xsd:boolean }?,

  ## default value: false
  attribute firstRow { xsd:boolean }?,

  ## default value: false
  attribute firstCol { xsd:boolean }?,

  ## default value: false
  attribute lastRow { xsd:boolean }?,

  ## default value: false
  attribute lastCol { xsd:boolean }?,

  ## default value: false
  attribute bandRow { xsd:boolean }?,

  ## default value: false
  attribute bandCol { xsd:boolean }?,
  a_EG_FillProperties?,
  a_EG_EffectProperties?,
  (element tableStyle { a_CT_TableStyle }
   | element tableStyleId { s_ST_Guid })?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_Table =
  element tblPr { a_CT_TableProperties }?,
  element tblGrid { a_CT_TableGrid },
  element tr { a_CT_TableRow }*
a_tbl = element tbl { a_CT_Table }
a_CT_Cell3D =

```



```
  ## default value: plastic
  attribute prstMaterial { a_ST_PresetMaterialType }?,
  element bevel { a_CT_Bevel },
  element lightRig { a_CT_LightRig }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_EG_ThemeableFillStyle =
  element fill { a_CT_FillProperties }
  | element fillRef { a_CT_StyleMatrixReference }
a_CT_ThemeableLineStyle =
  element ln { a_CT_LineProperties }
  | element lnRef { a_CT_StyleMatrixReference }
a_EG_ThemeableEffectStyle =
  element effect { a_CT_EffectProperties }
  | element effectRef { a_CT_StyleMatrixReference }
a_EG_ThemeableFontStyles =
  element font { a_CT_FontCollection }
  | element fontRef { a_CT_FontReference }
a_ST_OnOffStyleType = "on" | "off" | "def"
a_CT_TableStyleTextStyle =

  ## default value: def
  attribute b { a_ST_OnOffStyleType }?,

  ## default value: def
  attribute i { a_ST_OnOffStyleType }?,
  a_EG_ThemeableFontStyles?,
  a_EG_ColorChoice?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_TableCellBorderStyle =
  element left { a_CT_ThemeableLineStyle }?,
  element right { a_CT_ThemeableLineStyle }?,
  element top { a_CT_ThemeableLineStyle }?,
  element bottom { a_CT_ThemeableLineStyle }?,
  element insideH { a_CT_ThemeableLineStyle }?,
  element insideV { a_CT_ThemeableLineStyle }?,
  element tl2br { a_CT_ThemeableLineStyle }?,
  element tr2bl { a_CT_ThemeableLineStyle }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_TableBackgroundStyle =
  a_EG_ThemeableFillStyle?, a_EG_ThemeableEffectStyle?
a_CT_TableStyleCellStyle =
  element tcBdr { a_CT_TableCellBorderStyle }?,
  a_EG_ThemeableFillStyle?,
  element cell3D { a_CT_Cell3D }?
a_CT_TablePartStyle =
  element tcTxStyle { a_CT_TableStyleTextStyle }?,
  element tcStyle { a_CT_TableStyleCellStyle }?
a_CT_TableStyle =
  attribute styleId { s_ST_Guid },
  attribute styleName { xsd:string },
  element tblBg { a_CT_TableBackgroundStyle }?,
  element wholeTbl { a_CT_TablePartStyle }?,
  element band1H { a_CT_TablePartStyle }?,
```

```
  element band2H { a_CT_TablePartStyle }?,
  element band1V { a_CT_TablePartStyle }?,
  element band2V { a_CT_TablePartStyle }?,
  element lastCol { a_CT_TablePartStyle }?,
  element firstCol { a_CT_TablePartStyle }?,
  element lastRow { a_CT_TablePartStyle }?,
  element seCell { a_CT_TablePartStyle }?,
  element swCell { a_CT_TablePartStyle }?,
  element firstRow { a_CT_TablePartStyle }?,
  element neCell { a_CT_TablePartStyle }?,
  element nwCell { a_CT_TablePartStyle }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_TableStyleList =
  attribute def { s_ST_Guid },
  element tblStyle { a_CT_TableStyle }*
a_tblStyleLst = element tblStyleLst { a_CT_TableStyleList }
a_CT_TextParagraph =
  element pPr { a_CT_TextParagraphProperties }?,
  a_EG_TextRun*,
  element endParaRPr { a_CT_TextCharacterProperties }?
a_ST_TextAnchoringType = "t" | "ctr" | "b" | "just" | "dist"
a_ST_TextVertOverflowType = "overflow" | "ellipsis" | "clip"
a_ST_TextHorzOverflowType = "overflow" | "clip"
a_ST_TextVerticalType =
  "horz"
  | "vert"
  | "vert270"
  | "wordArtVert"
  | "eaVert"
  | "mongolianVert"
  | "wordArtVertRtl"
a_ST_TextWrappingType = "none" | "square"
a_ST_TextColumnCount =
  xsd:int { minInclusive = "1" maxInclusive = "16" }
a_CT_TextListStyle =
  element defPPr { a_CT_TextParagraphProperties }?,
  element lvl1pPr { a_CT_TextParagraphProperties }?,
  element lvl2pPr { a_CT_TextParagraphProperties }?,
  element lvl3pPr { a_CT_TextParagraphProperties }?,
  element lvl4pPr { a_CT_TextParagraphProperties }?,
  element lvl5pPr { a_CT_TextParagraphProperties }?,
  element lvl6pPr { a_CT_TextParagraphProperties }?,
  element lvl7pPr { a_CT_TextParagraphProperties }?,
  element lvl8pPr { a_CT_TextParagraphProperties }?,
  element lvl9pPr { a_CT_TextParagraphProperties }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_ST_TextFontScalePercentOrPercentString =
  a_ST_TextFontScalePercent | s_ST_Percentage
a_ST_TextFontScalePercent =
  xsd:int { minInclusive = "1000" maxInclusive = "100000" }
a_CT_TextNormalAutofit =

  ## default value: 100%
```



```
  attribute fontScale { a_ST_TextFontScalePercentOrPercentString }?,

  ## default value: 0%
  attribute lnSpcReduction { a_ST_TextSpacingPercentOrPercentString }?
a_CT_TextShapeAutofit = empty
a_CT_TextNoAutofit = empty
a_EG_TextAutofit =
  element noAutofit { a_CT_TextNoAutofit }
  | element normAutofit { a_CT_TextNormalAutofit }
  | element spAutoFit { a_CT_TextShapeAutofit }
a_CT_TextBodyProperties =
  attribute rot { a_ST_Angle }?,
  attribute spcFirstLastPara { xsd:boolean }?,
  attribute vertOverflow { a_ST_TextVertOverflowType }?,
  attribute horzOverflow { a_ST_TextHorzOverflowType }?,
  attribute vert { a_ST_TextVerticalType }?,
  attribute wrap { a_ST_TextWrappingType }?,
  attribute lIns { a_ST_Coordinate32 }?,
  attribute tIns { a_ST_Coordinate32 }?,
  attribute rIns { a_ST_Coordinate32 }?,
  attribute bIns { a_ST_Coordinate32 }?,
  attribute numCol { a_ST_TextColumnCount }?,
  attribute spcCol { a_ST_PositiveCoordinate32 }?,
  attribute rtlCol { xsd:boolean }?,
  attribute fromWordArt { xsd:boolean }?,
  attribute anchor { a_ST_TextAnchoringType }?,
  attribute anchorCtr { xsd:boolean }?,
  attribute forceAA { xsd:boolean }?,

  ## default value: false
  attribute upright { xsd:boolean }?,
  attribute compatLnSpc { xsd:boolean }?,
  element prstTxWarp { a_CT_PresetTextShape }?,
  a_EG_TextAutofit?,
  element scene3d { a_CT_Scene3D }?,
  a_EG_Text3D?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_TextBody =
  element bodyPr { a_CT_TextBodyProperties },
  element lstStyle { a_CT_TextListStyle }?,
  element p { a_CT_TextParagraph }+
a_ST_TextBulletStartAtNum =
  xsd:int { minInclusive = "1" maxInclusive = "32767" }
a_ST_TextAutonumberScheme =
  "alphaLcParenBoth"
  | "alphaUcParenBoth"
  | "alphaLcParenR"
  | "alphaUcParenR"
  | "alphaLcPeriod"
  | "alphaUcPeriod"
  | "arabicParenBoth"
  | "arabicParenR"
  | "arabicPeriod"
```

```
  | "arabicPlain"
  | "romanLcParenBoth"
  | "romanUcParenBoth"
  | "romanLcParenR"
  | "romanUcParenR"
  | "romanLcPeriod"
  | "romanUcPeriod"
  | "circleNumDbPlain"
  | "circleNumWdBlackPlain"
  | "circleNumWdWhitePlain"
  | "arabicDbPeriod"
  | "arabicDbPlain"
  | "ea1ChsPeriod"
  | "ea1ChsPlain"
  | "ea1ChtPeriod"
  | "ea1ChtPlain"
  | "ea1JpnChsDbPeriod"
  | "ea1JpnKorPlain"
  | "ea1JpnKorPeriod"
  | "arabic1Minus"
  | "arabic2Minus"
  | "hebrew2Minus"
  | "thaiAlphaPeriod"
  | "thaiAlphaParenR"
  | "thaiAlphaParenBoth"
  | "thaiNumPeriod"
  | "thaiNumParenR"
  | "thaiNumParenBoth"
  | "hindiAlphaPeriod"
  | "hindiNumPeriod"
  | "hindiNumParenR"
  | "hindiAlpha1Period"
a_CT_TextBulletColorFollowText = empty
a_EG_TextBulletColor =
  element buClrTx { a_CT_TextBulletColorFollowText }
  | element buClr { a_CT_Color }
a_ST_TextBulletSize = a_ST_TextBulletSizePercent | a_ST_TextBulletSizeDecimal
a_ST_TextBulletSizePercent =
  xsd:string {
    pattern = "0*((2[5-9])|([3-9][0-9])|([1-3][0-9][0-9])|400)%"
  }
a_ST_TextBulletSizeDecimal = xsd:int { minInclusive = "25000" maxInclusive = "400000" }
a_CT_TextBulletSizeFollowText = empty
a_CT_TextBulletSizePercent =
  attribute val { a_ST_TextBulletSizePercent }
a_CT_TextBulletSizePoint = attribute val { a_ST_TextFontSize }
a_EG_TextBulletSize =
  element buSzTx { a_CT_TextBulletSizeFollowText }
  | element buSzPct { a_CT_TextBulletSizePercent }
  | element buSzPts { a_CT_TextBulletSizePoint }
a_CT_TextBulletTypefaceFollowText = empty
a_EG_TextBulletTypeface =
  element buFontTx { a_CT_TextBulletTypefaceFollowText }
```



```
  | element buFont { a_CT_TextFont }
a_CT_TextAutonumberBullet =
  attribute type { a_ST_TextAutonumberScheme },

  ## default value: 1
  attribute startAt { a_ST_TextBulletStartAtNum }?
a_CT_TextCharBullet = attribute char { xsd:string }
a_CT_TextBlipBullet = element blip { a_CT_Blip }
a_CT_TextNoBullet = empty
a_EG_TextBullet =
  element buNone { a_CT_TextNoBullet }
  | element buAutoNum { a_CT_TextAutonumberBullet }
  | element buChar { a_CT_TextCharBullet }
  | element buBlip { a_CT_TextBlipBullet }
a_ST_TextPoint = a_ST_TextPointUnqualified | s_ST_UniversalMeasure
a_ST_TextPointUnqualified =
  xsd:int { minInclusive = "-400000" maxInclusive = "400000" }
a_ST_TextNonNegativePoint =
  xsd:int { minInclusive = "0" maxInclusive = "400000" }
a_ST_TextFontSize =
  xsd:int { minInclusive = "100" maxInclusive = "400000" }
a_ST_TextTypeface = xsd:string
a_ST_PitchFamily =
  xsd:byte "00" | xsd:byte "01" | xsd:byte "02" | xsd:byte "16" |
  xsd:byte "17" | xsd:byte "18" | xsd:byte "32" | xsd:byte "33" |
  xsd:byte "34" | xsd:byte "48" | xsd:byte "49" | xsd:byte "50" |
  xsd:byte "64" | xsd:byte "65" | xsd:byte "66" | xsd:byte "80" |
  xsd:byte "81" | xsd:byte "82"
a_CT_TextFont =
  attribute typeface { a_ST_TextTypeface },
  attribute panose { s_ST_Panose }?,

  ## default value: 0
  attribute pitchFamily { a_ST_PitchFamily }?,

  ## default value: 1
  attribute charset { xsd:byte }?
a_ST_TextUnderlineType =
  "none"
  | "words"
  | "sng"
  | "dbl"
  | "heavy"
  | "dotted"
  | "dottedHeavy"
  | "dash"
  | "dashHeavy"
  | "dashLong"
  | "dashLongHeavy"
  | "dotDash"
  | "dotDashHeavy"
  | "dotDotDash"
  | "dotDotDashHeavy"
```

```
  | "wavy"
  | "wavyHeavy"
  | "wavyDbl"
a_CT_TextUnderlineLineFollowText = empty
a_CT_TextUnderlineFillFollowText = empty
a_CT_TextUnderlineFillGroupWrapper = a_EG_FillProperties
a_EG_TextUnderlineLine =
  element uLnTx { a_CT_TextUnderlineLineFollowText }
  | element uLn { a_CT_LineProperties }?
a_EG_TextUnderlineFill =
  element uFillTx { a_CT_TextUnderlineFillFollowText }
  | element uFill { a_CT_TextUnderlineFillGroupWrapper }
a_ST_TextStrikeType = "noStrike" | "sngStrike" | "dblStrike"
a_ST_TextCapsType = "none" | "small" | "all"
a_CT_TextCharacterProperties =
  attribute kumimoji { xsd:boolean }?,
  attribute lang { s_ST_Lang }?,
  attribute altLang { s_ST_Lang }?,
  attribute sz { a_ST_TextFontSize }?,
  attribute b { xsd:boolean }?,
  attribute i { xsd:boolean }?,
  attribute u { a_ST_TextUnderlineType }?,
  attribute strike { a_ST_TextStrikeType }?,
  attribute kern { a_ST_TextNonNegativePoint }?,
  attribute cap { a_ST_TextCapsType }?,
  attribute spc { a_ST_TextPoint }?,
  attribute normalizeH { xsd:boolean }?,
  attribute baseline { a_ST_Percentage }?,
  attribute noProof { xsd:boolean }?,

  ## default value: true
  attribute dirty { xsd:boolean }?,

  ## default value: false
  attribute err { xsd:boolean }?,

  ## default value: true
  attribute smtClean { xsd:boolean }?,

  ## default value: 0
  attribute smtId { xsd:unsignedInt }?,
  attribute bmk { xsd:string }?,
  element ln { a_CT_LineProperties }?,
  a_EG_FillProperties?,
  a_EG_EffectProperties?,
  element highlight { a_CT_Color }?,
  a_EG_TextUnderlineLine?,
  a_EG_TextUnderlineFill?,
  element latin { a_CT_TextFont }?,
  element ea { a_CT_TextFont }?,
  element cs { a_CT_TextFont }?,
  element sym { a_CT_TextFont }?,
  element hlinkClick { a_CT_Hyperlink }?,
```



```
  element hlinkMouseOver { a_CT_Hyperlink }?,
  element rtl { a_CT_Boolean }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_Boolean =

  ## default value: 0
  attribute val { s_ST_OnOff }?
a_ST_TextSpacingPoint =
  xsd:int { minInclusive = "0" maxInclusive = "158400" }
a_ST_TextSpacingPercentOrPercentString =
  a_ST_TextSpacingPercent | s_ST_Percentage
a_ST_TextSpacingPercent =
  xsd:int { minInclusive = "0" maxInclusive = "13200000" }
a_CT_TextSpacingPercent =
  attribute val { a_ST_TextSpacingPercentOrPercentString }
a_CT_TextSpacingPoint = attribute val { a_ST_TextSpacingPoint }
a_ST_TextMargin =
  xsd:int { minInclusive = "0" maxInclusive = "51206400" }
a_ST_TextIndent =
  xsd:int { minInclusive = "-51206400" maxInclusive = "51206400" }
a_ST_TextTabAlignType = "l" | "ctr" | "r" | "dec"
a_CT_TextTabStop =
  attribute pos { a_ST_Coordinate32 }?,
  attribute algn { a_ST_TextTabAlignType }?
a_CT_TextTabStopList = element tab { a_CT_TextTabStop }*
a_CT_TextLineBreak = element rPr { a_CT_TextCharacterProperties }?
a_CT_TextSpacing =
  element spcPct { a_CT_TextSpacingPercent }
  | element spcPts { a_CT_TextSpacingPoint }
a_ST_TextAlignType =
  "l" | "ctr" | "r" | "just" | "justLow" | "dist" | "thaiDist"
a_ST_TextFontAlignType = "auto" | "t" | "ctr" | "base" | "b"
a_ST_TextIndentLevelType =
  xsd:int { minInclusive = "0" maxInclusive = "8" }
a_CT_TextParagraphProperties =
  attribute marL { a_ST_TextMargin }?,
  attribute marR { a_ST_TextMargin }?,
  attribute lvl { a_ST_TextIndentLevelType }?,
  attribute indent { a_ST_TextIndent }?,
  attribute algn { a_ST_TextAlignType }?,
  attribute defTabSz { a_ST_Coordinate32 }?,
  attribute rtl { xsd:boolean }?,
  attribute eaLnBrk { xsd:boolean }?,
  attribute fontAlgn { a_ST_TextFontAlignType }?,
  attribute latinLnBrk { xsd:boolean }?,
  attribute hangingPunct { xsd:boolean }?,
  element lnSpc { a_CT_TextSpacing }?,
  element spcBef { a_CT_TextSpacing }?,
  element spcAft { a_CT_TextSpacing }?,
  a_EG_TextBulletColor?,
  a_EG_TextBulletSize?,
  a_EG_TextBulletTypeface?,
  a_EG_TextBullet?,
```

```
  element tabLst { a_CT_TextTabStopList }?,
  element defRPr { a_CT_TextCharacterProperties }?,
  element extLst { a_CT_OfficeArtExtensionList }?
a_CT_TextField =
  attribute id { s_ST_Guid },
  attribute type { xsd:string }?,
  element rPr { a_CT_TextCharacterProperties }?,
  element pPr { a_CT_TextParagraphProperties }?,
  element t { xsd:string }?
a_EG_TextRun =
  element r { a_CT_RegularTextRun }
  | element br { a_CT_TextLineBreak }
  | element fld { a_CT_TextField }
a_CT_RegularTextRun =
  element rPr { a_CT_TextCharacterProperties }?,
  element t { xsd:string }
```

B.4.1.1 Part Schemas

B.4.1.1.1 Table Styles Part

This schema is available in the file DrawingML_Table_Styles.rnc.

```
include "dml-main.rnc"
include "shared-relationshipReference.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = a_tblStyleLst
```

B.4.1.1.2 Theme Part

This schema is available in the file DrawingML_Theme.rnc.

```
include "dml-main.rnc"
include "shared-relationshipReference.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = a_theme
```

B.4.1.1.3 Theme Override Part

This schema is available in the file DrawingML_Theme_Override.rnc.

```
include "dml-main.rnc"
```



```
include "shared-relationshipReference.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = a_themeOverride
```

B.4.2 DrawingML - Picture

This schema is available in the file dml-picture.rnc.

```
default namespace =
  "http://schemas.openxmlformats.org/drawingml/2006/picture"
namespace a = "http://schemas.openxmlformats.org/drawingml/2006/main"
namespace dpct =
  "http://schemas.openxmlformats.org/drawingml/2006/picture"
namespace o = "urn:schemas-microsoft-com:office:office"
namespace v = "urn:schemas-microsoft-com:vml"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"

dpct_CT_PictureNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvPicPr { a_CT_NonVisualPictureProperties }
dpct_CT_Picture =
  element nvPicPr { dpct_CT_PictureNonVisual },
  element blipFill { a_CT_BlipFillProperties },
  element spPr { a_CT_ShapeProperties }
dpct_pic = element pic { dpct_CT_Picture }
```

B.4.3 DrawingML - Locked Canvas

This schema is available in the file dml-lockedCanvas.rnc.

```
default namespace =
  "http://schemas.openxmlformats.org/drawingml/2006/lockedCanvas"
namespace a = "http://schemas.openxmlformats.org/drawingml/2006/main"
namespace dlckcnv =
  "http://schemas.openxmlformats.org/drawingml/2006/lockedCanvas"
namespace o = "urn:schemas-microsoft-com:office:office"
namespace r =
  "http://schemas.openxmlformats.org/officeDocument/2006/relationships"
namespace v = "urn:schemas-microsoft-com:vml"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"

dlckcnv_lockedCanvas = element lockedCanvas { a_CT_GvmlGroupShape }
```

B.4.4 DrawingML - Wordprocessing Drawing

This schema is available in the file dml-wordprocessingDrawing.rnc.

```
default namespace =
  "http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing"
namespace a = "http://schemas.openxmlformats.org/drawingml/2006/main"
namespace o = "urn:schemas-microsoft-com:office:office"
namespace dpct = "http://schemas.openxmlformats.org/drawingml/2006/picture"
namespace r = http://schemas.openxmlformats.org/officeDocument/2006/relationships
namespace v = "urn:schemas-microsoft-com:vml"
namespace w =
  "http://schemas.openxmlformats.org/wordprocessingml/2006/main"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace wp =
  "http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing"
namespace x = "urn:schemas-microsoft-com:office:excel"

wp_CT_EffectExtent =
  attribute l { a_ST_Coordinate },
  attribute t { a_ST_Coordinate },
  attribute r { a_ST_Coordinate },
  attribute b { a_ST_Coordinate }
wp_ST_WrapDistance = xsd:unsignedInt
wp_CT_Inline =
  attribute distT { wp_ST_WrapDistance }?,
  attribute distB { wp_ST_WrapDistance }?,
  attribute distL { wp_ST_WrapDistance }?,
  attribute distR { wp_ST_WrapDistance }?,
  element extent { a_CT_PositiveSize2D },
  element effectExtent { wp_CT_EffectExtent }?,
  element docPr { a_CT_NonVisualDrawingProps },
  element cNvGraphicFramePr { a_CT_NonVisualGraphicFrameProperties }?,
  a_graphic
wp_ST_WrapText = "bothSides" | "left" | "right" | "largest"
wp_CT_WrapPath =
  attribute edited { xsd:boolean }?,
  element start { a_CT_Point2D },
  element lineTo { a_CT_Point2D }+
wp_CT_WrapNone = empty
wp_CT_WrapSquare =
  attribute wrapText { wp_ST_WrapText },
  attribute distT { wp_ST_WrapDistance }?,
  attribute distB { wp_ST_WrapDistance }?,
  attribute distL { wp_ST_WrapDistance }?,
  attribute distR { wp_ST_WrapDistance }?,
  element effectExtent { wp_CT_EffectExtent }?
wp_CT_WrapTight =
  attribute wrapText { wp_ST_WrapText },
  attribute distL { wp_ST_WrapDistance }?,
  attribute distR { wp_ST_WrapDistance }?,
  element wrapPolygon { wp_CT_WrapPath }
wp_CT_WrapThrough =
  attribute wrapText { wp_ST_WrapText },
  attribute distL { wp_ST_WrapDistance }?,
  attribute distR { wp_ST_WrapDistance }?,
  element wrapPolygon { wp_CT_WrapPath }
```



```
wp_CT_WrapTopBottom =
  attribute distT { wp_ST_WrapDistance }?,
  attribute distB { wp_ST_WrapDistance }?,
  element effectExtent { wp_CT_EffectExtent }?
wp_EG_WrapType =
  element wrapNone { wp_CT_WrapNone }
  | element wrapSquare { wp_CT_WrapSquare }
  | element wrapTight { wp_CT_WrapTight }
  | element wrapThrough { wp_CT_WrapThrough }
  | element wrapTopAndBottom { wp_CT_WrapTopBottom }
wp_ST_PositionOffset = xsd:int
wp_ST_AlignH = "left" | "right" | "center" | "inside" | "outside"
wp_ST_RelFromH =
  "margin"
  | "page"
  | "column"
  | "character"
  | "leftMargin"
  | "rightMargin"
  | "insideMargin"
  | "outsideMargin"
wp_CT_PosH =
  attribute relativeFrom { wp_ST_RelFromH },
  (element align { wp_ST_AlignH }
   | element posOffset { wp_ST_PositionOffset })
wp_ST_AlignV = "top" | "bottom" | "center" | "inside" | "outside"
wp_ST_RelFromV =
  "margin"
  | "page"
  | "paragraph"
  | "line"
  | "topMargin"
  | "bottomMargin"
  | "insideMargin"
  | "outsideMargin"
wp_CT_PosV =
  attribute relativeFrom { wp_ST_RelFromV },
  (element align { wp_ST_AlignV }
   | element posOffset { wp_ST_PositionOffset })
wp_CT_Anchor =
  attribute distT { wp_ST_WrapDistance }?,
  attribute distB { wp_ST_WrapDistance }?,
  attribute distL { wp_ST_WrapDistance }?,
  attribute distR { wp_ST_WrapDistance }?,
  attribute simplePos { xsd:boolean }?,
  attribute relativeHeight { xsd:unsignedInt },
  attribute behindDoc { xsd:boolean },
  attribute locked { xsd:boolean },
  attribute layoutInCell { xsd:boolean },
  attribute hidden { xsd:boolean }?,
  attribute allowOverlap { xsd:boolean },
  element simplePos { a_CT_Point2D },
  element positionH { wp_CT_PosH },
```

```
  element positionV { wp_CT_PosV },
  element extent { a_CT_PositiveSize2D },
  element effectExtent { wp_CT_EffectExtent }?,
  wp_EG_WrapType,
  element docPr { a_CT_NonVisualDrawingProps },
  element cNvGraphicFramePr { a_CT_NonVisualGraphicFrameProperties }?,
  a_graphic
wp_CT_TxbxContent = w_EG_BlockLevelElts+
wp_CT_TextboxInfo =

  ## default value: 0
  attribute id { xsd:unsignedShort }?,
  element txbxContent { wp_CT_TxbxContent },
  element extLst { a_CT_OfficeArtExtensionList }?
wp_CT_LinkedTextboxInformation =
  attribute id { xsd:unsignedShort },
  attribute seq { xsd:unsignedShort },
  element extLst { a_CT_OfficeArtExtensionList }?
wp_CT_WordprocessingShape =

  ## default value: false
  attribute normalEastAsianFlow { xsd:boolean }?,
  element cNvPr { a_CT_NonVisualDrawingProps }?,
  (element cNvSpPr { a_CT_NonVisualDrawingShapeProps }
   | element cNvCnPr { a_CT_NonVisualConnectorProperties }),
  element spPr { a_CT_ShapeProperties },
  element style { a_CT_ShapeStyle }?,
  element extLst { a_CT_OfficeArtExtensionList }?,
  (element txbx { wp_CT_TextboxInfo }
   | element linkedTxbx { wp_CT_LinkedTextboxInformation })?,
  element bodyPr { a_CT_TextBodyProperties }
wp_CT_GraphicFrame =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvFrPr { a_CT_NonVisualGraphicFrameProperties },
  element xfrm { a_CT_Transform2D },
  a_graphic,
  element extLst { a_CT_OfficeArtExtensionList }?
wp_CT_WordprocessingContentPartNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps }?,
  element cNvContentPartPr { a_CT_NonVisualContentPartProperties }?
wp_CT_WordprocessingContentPart =
  attribute bwMode { a_ST_BlackWhiteMode }?,
  r_id,
  element nvContentPartPr { wp_CT_WordprocessingContentPartNonVisual }?,
  element xfrm { a_CT_Transform2D }?,
  element extLst { a_CT_OfficeArtExtensionList }?
wp_CT_WordprocessingGroup =
  element cNvPr { a_CT_NonVisualDrawingProps }?,
  element cNvGrpSpPr { a_CT_NonVisualGroupDrawingShapeProps },
  element grpSpPr { a_CT_GroupShapeProperties },
  (wp_wsp
   | element grpSp { wp_CT_WordprocessingGroup }
   | element graphicFrame { wp_CT_GraphicFrame }
```



```
    | dpct_pic
    | element contentPart { wp_CT_WordprocessingContentPart })*,
  element extLst { a_CT_OfficeArtExtensionList }?
wp_CT_WordprocessingCanvas =
  element bg { a_CT_BackgroundFormatting }?,
  element whole { a_CT_WholeE2oFormatting }?,
  (wp_wsp
    | dpct_pic
    | element contentPart { wp_CT_WordprocessingContentPart }
    | wp_wgp
    | element graphicFrame { wp_CT_GraphicFrame })*,
  element extLst { a_CT_OfficeArtExtensionList }?
wp_wpc = element wpc { wp_CT_WordprocessingCanvas }
wp_wgp = element wgp { wp_CT_WordprocessingGroup }
wp_wsp = element wsp { wp_CT_WordprocessingShape }
wp_inline = element inline { wp_CT_Inline }
wp_anchor = element anchor { wp_CT_Anchor }
```

B.4.5 DrawingML - Spreadsheet Drawing

This schema is available in the file dml-spreadsheetDrawing.rnc.

```
default namespace =
  "http://schemas.openxmlformats.org/drawingml/2006/spreadsheetDrawing"
namespace a = "http://schemas.openxmlformats.org/drawingml/2006/main"
namespace o = "urn:schemas-microsoft-com:office:office"
namespace r =
  "http://schemas.openxmlformats.org/officeDocument/2006/relationships"
namespace v = "urn:schemas-microsoft-com:vml"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"
namespace xdr =
  "http://schemas.openxmlformats.org/drawingml/2006/spreadsheetDrawing"

xdr_from = element from { xdr_CT_Marker }
xdr_to = element to { xdr_CT_Marker }
xdr_CT_AnchorClientData =

  ## default value: true
  attribute fLocksWithSheet { xsd:boolean }?,

  ## default value: true
  attribute fPrintsWithSheet { xsd:boolean }?
xdr_CT_ShapeNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvSpPr { a_CT_NonVisualDrawingShapeProps }
xdr_CT_Shape =
  attribute macro { xsd:string }?,
  attribute textlink { xsd:string }?,

  ## default value: true
  attribute fLocksText { xsd:boolean }?,

```

```
  ## default value: false
  attribute fPublished { xsd:boolean }?,
  element nvSpPr { xdr_CT_ShapeNonVisual },
  element spPr { a_CT_ShapeProperties },
  element style { a_CT_ShapeStyle }?,
  element txBody { a_CT_TextBody }?
xdr_CT_ConnectorNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvCxnSpPr { a_CT_NonVisualConnectorProperties }
xdr_CT_Connector =
  attribute macro { xsd:string }?,

  ## default value: false
  attribute fPublished { xsd:boolean }?,
  element nvCxnSpPr { xdr_CT_ConnectorNonVisual },
  element spPr { a_CT_ShapeProperties },
  element style { a_CT_ShapeStyle }?
xdr_CT_PictureNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvPicPr { a_CT_NonVisualPictureProperties }
xdr_CT_Picture =
  attribute macro { xsd:string }?,

  ## default value: false
  attribute fPublished { xsd:boolean }?,
  element nvPicPr { xdr_CT_PictureNonVisual },
  element blipFill { a_CT_BlipFillProperties },
  element spPr { a_CT_ShapeProperties },
  element style { a_CT_ShapeStyle }?
xdr_CT_GraphicalObjectFrameNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvGraphicFramePr { a_CT_NonVisualGraphicFrameProperties }
xdr_CT_GraphicalObjectFrame =
  attribute macro { xsd:string }?,

  ## default value: false
  attribute fPublished { xsd:boolean }?,
  element nvGraphicFramePr { xdr_CT_GraphicalObjectFrameNonVisual },
  element xfrm { a_CT_Transform2D },
  a_graphic
xdr_CT_GroupShapeNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvGrpSpPr { a_CT_NonVisualGroupDrawingShapeProps }
xdr_CT_GroupShape =
  element nvGrpSpPr { xdr_CT_GroupShapeNonVisual },
  element grpSpPr { a_CT_GroupShapeProperties },
  (element sp { xdr_CT_Shape }
   | element grpSp { xdr_CT_GroupShape }
   | element graphicFrame { xdr_CT_GraphicalObjectFrame }
   | element cxnSp { xdr_CT_Connector }
   | element pic { xdr_CT_Picture })*
xdr_EG_ObjectChoices =
  element sp { xdr_CT_Shape }
```



```
  | element grpSp { xdr_CT_GroupShape }
  | element graphicFrame { xdr_CT_GraphicalObjectFrame }
  | element cxnSp { xdr_CT_Connector }
  | element pic { xdr_CT_Picture }
  | element contentPart { xdr_CT_Rel }
xdr_CT_Rel = r_id
xdr_ST_ColID = xsd:int { minInclusive = "0" }
xdr_ST_RowID = xsd:int { minInclusive = "0" }
xdr_CT_Marker =
  element col { xdr_ST_ColID },
  element colOff { a_ST_Coordinate },
  element row { xdr_ST_RowID },
  element rowOff { a_ST_Coordinate }
xdr_ST_EditAs = "twoCell" | "oneCell" | "absolute"
xdr_CT_TwoCellAnchor =

  ## default value: twoCell
  attribute editAs { xdr_ST_EditAs }?,
  element from { xdr_CT_Marker },
  element to { xdr_CT_Marker },
  xdr_EG_ObjectChoices,
  element clientData { xdr_CT_AnchorClientData }
xdr_CT_OneCellAnchor =
  element from { xdr_CT_Marker },
  element ext { a_CT_PositiveSize2D },
  xdr_EG_ObjectChoices,
  element clientData { xdr_CT_AnchorClientData }
xdr_CT_AbsoluteAnchor =
  element pos { a_CT_Point2D },
  element ext { a_CT_PositiveSize2D },
  xdr_EG_ObjectChoices,
  element clientData { xdr_CT_AnchorClientData }
xdr_EG_Anchor =
  element twoCellAnchor { xdr_CT_TwoCellAnchor }
  | element oneCellAnchor { xdr_CT_OneCellAnchor }
  | element absoluteAnchor { xdr_CT_AbsoluteAnchor }
xdr_CT_Drawing = xdr_EG_Anchor*
xdr_wsDr = element wsDr { xdr_CT_Drawing }
```

B.5 DrawingML - Components

B.5.1 DrawingML - Chart

This schema is available in the file dml-chart.rnc.

```
default namespace =
  "http://schemas.openxmlformats.org/drawingml/2006/chart"
namespace a = "http://schemas.openxmlformats.org/drawingml/2006/main"
namespace cdr =
  "http://schemas.openxmlformats.org/drawingml/2006/chartDrawing"
namespace dchrt =
  "http://schemas.openxmlformats.org/drawingml/2006/chart"
namespace o = "urn:schemas-microsoft-com:office:office"
```

```
namespace r =
  "http://schemas.openxmlformats.org/officeDocument/2006/relationships"
namespace s =
  "http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
namespace v = "urn:schemas-microsoft-com:vml"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"

dchrt_CT_Boolean =

  ## default value: true
  attribute val { xsd:boolean }?
dchrt_CT_Double = attribute val { xsd:double }
dchrt_CT_UnsignedInt = attribute val { xsd:unsignedInt }
dchrt_CT_RelId = r_id
dchrt_CT_Extension =
  attribute uri { xsd:token }?,
  dchrt_CT_Extension_any
dchrt_CT_Extension_any =
  element * - (o:* | v:* | w10:* | x:*) {
    anyAttribute*,
    mixed { anyElement* }
  }
dchrt_CT_ExtensionList = element ext { dchrt_CT_Extension }*
dchrt_CT_NumVal =
  attribute idx { xsd:unsignedInt },
  attribute formatCode { s_ST_Xstring }?,
  element v { s_ST_Xstring }
dchrt_CT_NumData =
  element formatCode { s_ST_Xstring }?,
  element ptCount { dchrt_CT_UnsignedInt }?,
  element pt { dchrt_CT_NumVal }*,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_NumRef =
  element f { xsd:string },
  element numCache { dchrt_CT_NumData }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_NumDataSource =
  element numRef { dchrt_CT_NumRef }
  | element numLit { dchrt_CT_NumData }
dchrt_CT_StrVal =
  attribute idx { xsd:unsignedInt },
  element v { s_ST_Xstring }
dchrt_CT_StrData =
  element ptCount { dchrt_CT_UnsignedInt }?,
  element pt { dchrt_CT_StrVal }*,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_StrRef =
  element f { xsd:string },
  element strCache { dchrt_CT_StrData }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_Tx =
  element strRef { dchrt_CT_StrRef }
```



```
  | element rich { a_CT_TextBody }
dchrt_CT_TextLanguageID = attribute val { s_ST_Lang }
dchrt_CT_Lvl = element pt { dchrt_CT_StrVal }*
dchrt_CT_MultiLvlStrData =
  element ptCount { dchrt_CT_UnsignedInt }?,
  element lvl { dchrt_CT_Lvl }*,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_MultiLvlStrRef =
  element f { xsd:string },
  element multiLvlStrCache { dchrt_CT_MultiLvlStrData }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_AxDataSource =
  element multiLvlStrRef { dchrt_CT_MultiLvlStrRef }
  | element numRef { dchrt_CT_NumRef }
  | element numLit { dchrt_CT_NumData }
  | element strRef { dchrt_CT_StrRef }
  | element strLit { dchrt_CT_StrData }
dchrt_CT_SerTx =
  element strRef { dchrt_CT_StrRef }
  | element v { s_ST_Xstring }
dchrt_ST_LayoutTarget = string "inner" | string "outer"
dchrt_CT_LayoutTarget =

  ## default value: outer
  attribute val { dchrt_ST_LayoutTarget }?
dchrt_ST_LayoutMode = string "edge" | string "factor"
dchrt_CT_LayoutMode =

  ## default value: factor
  attribute val { dchrt_ST_LayoutMode }?
dchrt_CT_ManualLayout =
  element layoutTarget { dchrt_CT_LayoutTarget }?,
  element xMode { dchrt_CT_LayoutMode }?,
  element yMode { dchrt_CT_LayoutMode }?,
  element wMode { dchrt_CT_LayoutMode }?,
  element hMode { dchrt_CT_LayoutMode }?,
  element x { dchrt_CT_Double }?,
  element y { dchrt_CT_Double }?,
  element w { dchrt_CT_Double }?,
  element h { dchrt_CT_Double }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_Layout =
  element manualLayout { dchrt_CT_ManualLayout }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_Title =
  element tx { dchrt_CT_Tx }?,
  element layout { dchrt_CT_Layout }?,
  element overlay { dchrt_CT_Boolean }?,
  element spPr { a_CT_ShapeProperties }?,
  element txPr { a_CT_TextBody }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_ST_RotX = xsd:byte { minInclusive = "-90" maxInclusive = "90" }
dchrt_CT_RotX =
```

```

  ## default value: 0
  attribute val { dchrt_ST_RotX }?
dchrt_ST_HPercent =
dchrt_ST_HPercentWithSymbol | dchrt_ST_HPercentUShort
dchrt_ST_HPercentWithSymbol =
  xsd:string {
    pattern = "0*(([5-9])|([1-9][0-9])|([1-4][0-9][0-9])|500)%"
  }
dchrt_ST_HPercentUShort =
  xsd:unsignedShort { minInclusive = "5" maxInclusive = "500" }
dchrt_CT_HPercent =

  ## default value: 100%
  attribute val { dchrt_ST_HPercent }?
dchrt_ST_RotY =
  xsd:unsignedShort { minInclusive = "0" maxInclusive = "360" }
dchrt_CT_RotY =

  ## default value: 0
  attribute val { dchrt_ST_RotY }?
dchrt_ST_DepthPercent =
dchrt_ST_DepthPercentWithSymbol | dchrt_ST_DepthPercentUShort
dchrt_ST_DepthPercentWithSymbol =
  xsd:string {
    pattern = "0*(([2-9][0-9])|([1-9][0-9][0-9])|(1[0-9][0-9][0-9])|2000)%"
  }
dchrt_ST_DepthPercentUShort =
  xsd:unsignedShort { minInclusive = "20" maxInclusive = "2000" }
dchrt_CT_DepthPercent =

  ## default value: 100%
  attribute val { dchrt_ST_DepthPercent }?
dchrt_ST_Perspective =
  xsd:unsignedByte { minInclusive = "0" maxInclusive = "240" }
dchrt_CT_Perspective =

  ## default value: 30
  attribute val { dchrt_ST_Perspective }?
dchrt_CT_View3D =
  element rotX { dchrt_CT_RotX }?,
  element hPercent { dchrt_CT_HPercent }?,
  element rotY { dchrt_CT_RotY }?,
  element depthPercent { dchrt_CT_DepthPercent }?,
  element rAngAx { dchrt_CT_Boolean }?,
  element perspective { dchrt_CT_Perspective }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_Surface =
  element thickness { dchrt_CT_Thickness}?,
  element spPr { a_CT_ShapeProperties }?,
  element pictureOptions { dchrt_CT_PictureOptions }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_ST_Thickness = dchrt_ST_ThicknessPercent | xsd:unsignedInt
```



```
dchrt_ST_ThicknessPercent = xsd:string { pattern = "([0-9]+)%" }
dchrt_CT_Thickness = attribute val { dchrt_ST_Thickness }
dchrt_CT_DTable =
  element showHorzBorder { dchrt_CT_Boolean }?,
  element showVertBorder { dchrt_CT_Boolean }?,
  element showOutline { dchrt_CT_Boolean }?,
  element showKeys { dchrt_CT_Boolean }?,
  element spPr { a_CT_ShapeProperties }?,
  element txPr { a_CT_TextBody }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_ST_GapAmount =
dchrt_ST_GapAmountPercent | dchrt_ST_GapAmountUShort
dchrt_ST_GapAmountPercent =
  xsd:string {
    pattern = "0*(([0-9])|([1-9][0-9])|([1-4][0-9][0-9])|500)%"
  }
dchrt_ST_GapAmountUShort =
  xsd:unsignedShort { minInclusive = "0" maxInclusive = "500" }
dchrt_CT_GapAmount =

  ## default value: 150%
  attribute val { dchrt_ST_GapAmount }?
dchrt_ST_Overlap =
  dchrt_ST_OverlapPercent | dchrt_ST_OverlapByte
dchrt_ST_OverlapPercent =
  xsd:string { pattern = "(-?0*(([0-9])|([1-9][0-9])|100))%" }
dchrt_ST_OverlapByte =
  xsd:byte { minInclusive = "-100" maxInclusive = "100" }
dchrt_CT_Overlap =

  ## default value: 0%
  attribute val { dchrt_ST_Overlap }?
dchrt_ST_BubbleScale =
  dchrt_ST_BubbleScalePercent | dchrt_ST_BubbleScaleUInt
dchrt_ST_BubbleScalePercent =
  xsd:string {
    pattern = "0*(([0-9])|([1-9][0-9])|([1-2][0-9][0-9])|300)%"
  }
dchrt_ST_BubbleScaleUInt =
  xsd:unsignedInt { minInclusive = "0" maxInclusive = "300" }
dchrt_CT_BubbleScale =

  ## default value: 100%
  attribute val { dchrt_ST_BubbleScale }?
dchrt_ST_SizeRepresents = string "area" | string "w"
dchrt_CT_SizeRepresents =

  ## default value: area
  attribute val { dchrt_ST_SizeRepresents }?
dchrt_ST_FirstSliceAng =
  xsd:unsignedShort { minInclusive = "0" maxInclusive = "360" }
dchrt_CT_FirstSliceAng =

```

```
  ## default value: 0
  attribute val { dchrt_ST_FirstSliceAng }?
dchrt_ST_HoleSize =
dchrt_ST_HoleSizePercent | dchrt_ST_HoleSizeUByte
dchrt_ST_HoleSizePercent = xsd:string { pattern = "0*([1-9]|([1-8][0-9])|90)%" }
dchrt_ST_HoleSizeUByte = xsd:unsignedByte { minInclusive = "1" maxInclusive = "90" }
dchrt_CT_HoleSize =

  ## default value: 10%
  attribute val { dchrt_ST_HoleSize }?
dchrt_ST_SplitType =
  string "auto"
  | string "cust"
  | string "percent"
  | string "pos"
  | string "val"
dchrt_CT_SplitType =

  ## default value: auto
  attribute val { dchrt_ST_SplitType }?
dchrt_CT_CustSplit = element secondPiePt { dchrt_CT_UnsignedInt }*
dchrt_ST_SecondPieSize =
dchrt_ST_SecondPieSizePercent | dchrt_ST_SecondPieSizeUShort
dchrt_ST_SecondPieSizePercent =
  xsd:string { pattern = "0*(([5-9])|([1-9][0-9])|(1[0-9][0-9])|200)%" }
dchrt_ST_SecondPieSizeUShort =
  xsd:unsignedShort { minInclusive = "5" maxInclusive = "200" }
dchrt_CT_SecondPieSize =

  ## default value: 75%
  attribute val { dchrt_ST_SecondPieSize }?
dchrt_CT_NumFmt =
  attribute formatCode { s_ST_Xstring },
  attribute sourceLinked { xsd:boolean }?
dchrt_ST_LblAlgn = string "ctr" | string "l" | string "r"
dchrt_CT_LblAlgn = attribute val { dchrt_ST_LblAlgn }
dchrt_ST_DLblPos =
  string "bestFit"
  | string "b"
  | string "ctr"
  | string "inBase"
  | string "inEnd"
  | string "l"
  | string "outEnd"
  | string "r"
  | string "t"
dchrt_CT_DLblPos = attribute val { dchrt_ST_DLblPos }
dchrt_EG_DLblShared =
  element numFmt { dchrt_CT_NumFmt }?,
  element spPr { a_CT_ShapeProperties }?,
  element txPr { a_CT_TextBody }?,
  element dLblPos { dchrt_CT_DLblPos }?,
  element showLegendKey { dchrt_CT_Boolean }?,
```



```
  element showVal { dchrt_CT_Boolean }?,
  element showCatName { dchrt_CT_Boolean }?,
  element showSerName { dchrt_CT_Boolean }?,
  element showPercent { dchrt_CT_Boolean }?,
  element showBubbleSize { dchrt_CT_Boolean }?,
  element separator { xsd:string }?
dchrt_Group_DLbl =
  element layout { dchrt_CT_Layout }?,
  element tx { dchrt_CT_Tx }?,
  dchrt_EG_DLblShared
dchrt_CT_DLbl =
  element idx { dchrt_CT_UnsignedInt },
  (element delete { dchrt_CT_Boolean }
   | dchrt_Group_DLbl),
  element extLst { dchrt_CT_ExtensionList }?
dchrt_Group_DLbls =
  dchrt_EG_DLblShared,
  element showLeaderLines { dchrt_CT_Boolean }?,
  element leaderLines { dchrt_CT_ChartLines }?
dchrt_CT_DLbls =
  element dLbl { dchrt_CT_DLbl }*,
  (element delete { dchrt_CT_Boolean }
   | dchrt_Group_DLbls),
  element extLst { dchrt_CT_ExtensionList }?
dchrt_ST_MarkerStyle =
  string "circle"
  | string "dash"
  | string "diamond"
  | string "dot"
  | string "none"
  | string "picture"
  | string "plus"
  | string "square"
  | string "star"
  | string "triangle"
  | string "x"
  | string "auto"
dchrt_CT_MarkerStyle = attribute val { dchrt_ST_MarkerStyle }
dchrt_ST_MarkerSize =
  xsd:unsignedByte { minInclusive = "2" maxInclusive = "72" }
dchrt_CT_MarkerSize =

  ## default value: 5
  attribute val { dchrt_ST_MarkerSize }?
dchrt_CT_Marker =
  element symbol { dchrt_CT_MarkerStyle }?,
  element size { dchrt_CT_MarkerSize }?,
  element spPr { a_CT_ShapeProperties }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_DPt =
  element idx { dchrt_CT_UnsignedInt },
  element invertIfNegative { dchrt_CT_Boolean }?,
  element marker { dchrt_CT_Marker }?,
```

```
  element bubble3D { dchrt_CT_Boolean }?,
  element explosion { dchrt_CT_UnsignedInt }?,
  element spPr { a_CT_ShapeProperties }?,
  element pictureOptions { dchrt_CT_PictureOptions }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_ST_TrendlineType =
  string "exp"
  | string "linear"
  | string "log"
  | string "movingAvg"
  | string "poly"
  | string "power"
dchrt_CT_TrendlineType =

  ## default value: linear
  attribute val { dchrt_ST_TrendlineType }?
dchrt_ST_Order =
  xsd:unsignedByte { minInclusive = "2" maxInclusive = "6" }
dchrt_CT_Order =

  ## default value: 2
  attribute val { dchrt_ST_Order }?
dchrt_ST_Period =
  xsd:unsignedInt { minInclusive = "2" }
dchrt_CT_Period =

  ## default value: 2
  attribute val { dchrt_ST_Period }?
dchrt_CT_TrendlineLbl =
  element layout { dchrt_CT_Layout }?,
  element tx { dchrt_CT_Tx }?,
  element numFmt { dchrt_CT_NumFmt }?,
  element spPr { a_CT_ShapeProperties }?,
  element txPr { a_CT_TextBody }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_Trendline =
  element name { xsd:string }?,
  element spPr { a_CT_ShapeProperties }?,
  element trendlineType { dchrt_CT_TrendlineType },
  element order { dchrt_CT_Order }?,
  element period { dchrt_CT_Period }?,
  element forward { dchrt_CT_Double }?,
  element backward { dchrt_CT_Double }?,
  element intercept { dchrt_CT_Double }?,
  element dispRSqr { dchrt_CT_Boolean }?,
  element dispEq { dchrt_CT_Boolean }?,
  element trendlineLbl { dchrt_CT_TrendlineLbl }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_ST_ErrDir = string "x" | string "y"
dchrt_CT_ErrDir = attribute val { dchrt_ST_ErrDir }
dchrt_ST_ErrBarType = string "both" | string "minus" | string "plus"
dchrt_CT_ErrBarType =

```



```
  ## default value: both
  attribute val { dchrt_ST_ErrBarType }?
dchrt_ST_ErrValType =
  string "cust"
  | string "fixedVal"
  | string "percentage"
  | string "stdDev"
  | string "stdErr"
dchrt_CT_ErrValType =

  ## default value: fixedVal
  attribute val { dchrt_ST_ErrValType }?
dchrt_CT_ErrBars =
  element errDir { dchrt_CT_ErrDir }?,
  element errBarType { dchrt_CT_ErrBarType },
  element errValType { dchrt_CT_ErrValType },
  element noEndCap { dchrt_CT_Boolean }?,
  element plus { dchrt_CT_NumDataSource }?,
  element minus { dchrt_CT_NumDataSource }?,
  element val { dchrt_CT_Double }?,
  element spPr { a_CT_ShapeProperties }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_UpDownBar = element spPr { a_CT_ShapeProperties }?
dchrt_CT_UpDownBars =
  element gapWidth { dchrt_CT_GapAmount }?,
  element upBars { dchrt_CT_UpDownBar }?,
  element downBars { dchrt_CT_UpDownBar }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_EG_SerShared =
  element idx { dchrt_CT_UnsignedInt },
  element order { dchrt_CT_UnsignedInt },
  element tx { dchrt_CT_SerTx }?,
  element spPr { a_CT_ShapeProperties }?
dchrt_CT_LineSer =
  dchrt_EG_SerShared,
  element marker { dchrt_CT_Marker }?,
  element dPt { dchrt_CT_DPt }*,
  element dLbls { dchrt_CT_DLbls }?,
  element trendline { dchrt_CT_Trendline }*,
  element errBars { dchrt_CT_ErrBars }?,
  element cat { dchrt_CT_AxDataSource }?,
  element val { dchrt_CT_NumDataSource }?,
  element smooth { dchrt_CT_Boolean }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_ScatterSer =
  dchrt_EG_SerShared,
  element marker { dchrt_CT_Marker }?,
  element dPt { dchrt_CT_DPt }*,
  element dLbls { dchrt_CT_DLbls }?,
  element trendline { dchrt_CT_Trendline }*,
  element errBars { dchrt_CT_ErrBars }*,
  element xVal { dchrt_CT_AxDataSource }?,
  element yVal { dchrt_CT_NumDataSource }?,
```

```
  element smooth { dchrt_CT_Boolean }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_RadarSer =
  dchrt_EG_SerShared,
  element marker { dchrt_CT_Marker }?,
  element dPt { dchrt_CT_DPt }*,
  element dLbls { dchrt_CT_DLbls }?,
  element cat { dchrt_CT_AxDataSource }?,
  element val { dchrt_CT_NumDataSource }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_BarSer =
  dchrt_EG_SerShared,
  element invertIfNegative { dchrt_CT_Boolean }?,
  element pictureOptions { dchrt_CT_PictureOptions }?,
  element dPt { dchrt_CT_DPt }*,
  element dLbls { dchrt_CT_DLbls }?,
  element trendline { dchrt_CT_Trendline }*,
  element errBars { dchrt_CT_ErrBars }?,
  element cat { dchrt_CT_AxDataSource }?,
  element val { dchrt_CT_NumDataSource }?,
  element shape { dchrt_CT_Shape }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_AreaSer =
  dchrt_EG_SerShared,
  element pictureOptions { dchrt_CT_PictureOptions }?,
  element dPt { dchrt_CT_DPt }*,
  element dLbls { dchrt_CT_DLbls }?,
  element trendline { dchrt_CT_Trendline }*,
  element errBars { dchrt_CT_ErrBars }*,
  element cat { dchrt_CT_AxDataSource }?,
  element val { dchrt_CT_NumDataSource }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_PieSer =
  dchrt_EG_SerShared,
  element explosion { dchrt_CT_UnsignedInt }?,
  element dPt { dchrt_CT_DPt }*,
  element dLbls { dchrt_CT_DLbls }?,
  element cat { dchrt_CT_AxDataSource }?,
  element val { dchrt_CT_NumDataSource }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_BubbleSer =
  dchrt_EG_SerShared,
  element invertIfNegative { dchrt_CT_Boolean }?,
  element dPt { dchrt_CT_DPt }*,
  element dLbls { dchrt_CT_DLbls }?,
  element trendline { dchrt_CT_Trendline }*,
  element errBars { dchrt_CT_ErrBars }*,
  element xVal { dchrt_CT_AxDataSource }?,
  element yVal { dchrt_CT_NumDataSource }?,
  element bubbleSize { dchrt_CT_NumDataSource }?,
  element bubble3D { dchrt_CT_Boolean }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_SurfaceSer =
```



```
  dchrt_EG_SerShared,
  element cat { dchrt_CT_AxDataSource }?,
  element val { dchrt_CT_NumDataSource }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_ST_Grouping =
  string "percentStacked" | string "standard" | string "stacked"
dchrt_CT_Grouping =

  ## default value: standard
  attribute val { dchrt_ST_Grouping }?
dchrt_CT_ChartLines = element spPr { a_CT_ShapeProperties }?
dchrt_EG_LineChartShared =
  element grouping { dchrt_CT_Grouping },
  element varyColors { dchrt_CT_Boolean }?,
  element ser { dchrt_CT_LineSer }*,
  element dLbls { dchrt_CT_DLbls }?,
  element dropLines { dchrt_CT_ChartLines }?
dchrt_CT_LineChart =
  dchrt_EG_LineChartShared,
  element hiLowLines { dchrt_CT_ChartLines }?,
  element upDownBars { dchrt_CT_UpDownBars }?,
  element marker { dchrt_CT_Boolean }?,
  element smooth { dchrt_CT_Boolean }?,
  element axId { dchrt_CT_UnsignedInt }+,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_Line3DChart =
  dchrt_EG_LineChartShared,
  element gapDepth { dchrt_CT_GapAmount }?,
  element axId { dchrt_CT_UnsignedInt }+,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_StockChart =
  element ser { dchrt_CT_LineSer }+,
  element dLbls { dchrt_CT_DLbls }?,
  element dropLines { dchrt_CT_ChartLines }?,
  element hiLowLines { dchrt_CT_ChartLines }?,
  element upDownBars { dchrt_CT_UpDownBars }?,
  element axId { dchrt_CT_UnsignedInt }+,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_ST_ScatterStyle =
  string "none"
  | string "line"
  | string "lineMarker"
  | string "marker"
  | string "smooth"
  | string "smoothMarker"
dchrt_CT_ScatterStyle =

  ## default value: marker
  attribute val { dchrt_ST_ScatterStyle }?
dchrt_CT_ScatterChart =
  element scatterStyle { dchrt_CT_ScatterStyle },
  element varyColors { dchrt_CT_Boolean }?,
  element ser { dchrt_CT_ScatterSer }*,
```

```
  element dLbls { dchrt_CT_DLbls }?,
  element axId { dchrt_CT_UnsignedInt }+,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_ST_RadarStyle =
  string "standard" | string "marker" | string "filled"
dchrt_CT_RadarStyle =

  ## default value: standard
  attribute val { dchrt_ST_RadarStyle }?
dchrt_CT_RadarChart =
  element radarStyle { dchrt_CT_RadarStyle },
  element varyColors { dchrt_CT_Boolean }?,
  element ser { dchrt_CT_RadarSer }*,
  element dLbls { dchrt_CT_DLbls }?,
  element axId { dchrt_CT_UnsignedInt }+,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_ST_BarGrouping =
  string "percentStacked"
  | string "clustered"
  | string "standard"
  | string "stacked"
dchrt_CT_BarGrouping =

  ## default value: clustered
  attribute val { dchrt_ST_BarGrouping }?
dchrt_ST_BarDir = string "bar" | string "col"
dchrt_CT_BarDir =

  ## default value: col
  attribute val { dchrt_ST_BarDir }?
dchrt_ST_Shape =
  string "cone"
  | string "coneToMax"
  | string "box"
  | string "cylinder"
  | string "pyramid"
  | string "pyramidToMax"
dchrt_CT_Shape =

  ## default value: box
  attribute val { dchrt_ST_Shape }?
dchrt_EG_BarChartShared =
  element barDir { dchrt_CT_BarDir },
  element grouping { dchrt_CT_BarGrouping }?,
  element varyColors { dchrt_CT_Boolean }?,
  element ser { dchrt_CT_BarSer }*,
  element dLbls { dchrt_CT_DLbls }?
dchrt_CT_BarChart =
  dchrt_EG_BarChartShared,
  element gapWidth { dchrt_CT_GapAmount }?,
  element overlap { dchrt_CT_Overlap }?,
  element serLines { dchrt_CT_ChartLines }*,
  element axId { dchrt_CT_UnsignedInt }+,
```



```
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_Bar3DChart =
  dchrt_EG_BarChartShared,
  element gapWidth { dchrt_CT_GapAmount }?,
  element gapDepth { dchrt_CT_GapAmount }?,
  element shape { dchrt_CT_Shape }?,
  element axId { dchrt_CT_UnsignedInt }+,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_EG_AreaChartShared =
  element grouping { dchrt_CT_Grouping }?,
  element varyColors { dchrt_CT_Boolean }?,
  element ser { dchrt_CT_AreaSer }*,
  element dLbls { dchrt_CT_DLbls }?,
  element dropLines { dchrt_CT_ChartLines }?
dchrt_CT_AreaChart =
  dchrt_EG_AreaChartShared,
  element axId { dchrt_CT_UnsignedInt }+,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_Area3DChart =
  dchrt_EG_AreaChartShared,
  element gapDepth { dchrt_CT_GapAmount }?,
  element axId { dchrt_CT_UnsignedInt }+,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_EG_PieChartShared =
  element varyColors { dchrt_CT_Boolean }?,
  element ser { dchrt_CT_PieSer }*,
  element dLbls { dchrt_CT_DLbls }?
dchrt_CT_PieChart =
  dchrt_EG_PieChartShared,
  element firstSliceAng { dchrt_CT_FirstSliceAng }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_Pie3DChart =
  dchrt_EG_PieChartShared,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_DoughnutChart =
  dchrt_EG_PieChartShared,
  element firstSliceAng { dchrt_CT_FirstSliceAng }?,
  element holeSize { dchrt_CT_HoleSize }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_ST_OfPieType = string "pie" | string "bar"
dchrt_CT_OfPieType =

  ## default value: pie
  attribute val { dchrt_ST_OfPieType }?
dchrt_CT_OfPieChart =
  element ofPieType { dchrt_CT_OfPieType },
  dchrt_EG_PieChartShared,
  element gapWidth { dchrt_CT_GapAmount }?,
  element splitType { dchrt_CT_SplitType }?,
  element splitPos { dchrt_CT_Double }?,
  element custSplit { dchrt_CT_CustSplit }?,
  element secondPieSize { dchrt_CT_SecondPieSize }?,
  element serLines { dchrt_CT_ChartLines }*,
```

```
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_BubbleChart =
  element varyColors { dchrt_CT_Boolean }?,
  element ser { dchrt_CT_BubbleSer }*,
  element dLbls { dchrt_CT_DLbls }?,
  element bubble3D { dchrt_CT_Boolean }?,
  element bubbleScale { dchrt_CT_BubbleScale }?,
  element showNegBubbles { dchrt_CT_Boolean }?,
  element sizeRepresents { dchrt_CT_SizeRepresents }?,
  element axId { dchrt_CT_UnsignedInt }+,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_BandFmt =
  element idx { dchrt_CT_UnsignedInt },
  element spPr { a_CT_ShapeProperties }?
dchrt_CT_BandFmts = element bandFmt { dchrt_CT_BandFmt }*
dchrt_EG_SurfaceChartShared =
  element wireframe { dchrt_CT_Boolean }?,
  element ser { dchrt_CT_SurfaceSer }*,
  element bandFmts { dchrt_CT_BandFmts }?
dchrt_CT_SurfaceChart =
  dchrt_EG_SurfaceChartShared,
  element axId { dchrt_CT_UnsignedInt }+,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_Surface3DChart =
  dchrt_EG_SurfaceChartShared,
  element axId { dchrt_CT_UnsignedInt }+,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_ST_AxPos = string "b" | string "l" | string "r" | string "t"
dchrt_CT_AxPos = attribute val { dchrt_ST_AxPos }
dchrt_ST_Crosses = string "autoZero" | string "max" | string "min"
dchrt_CT_Crosses = attribute val { dchrt_ST_Crosses }
dchrt_ST_CrossBetween = string "between" | string "midCat"
dchrt_CT_CrossBetween = attribute val { dchrt_ST_CrossBetween }
dchrt_ST_TickMark =
  string "cross" | string "in" | string "none" | string "out"
dchrt_CT_TickMark =

  ## default value: cross
  attribute val { dchrt_ST_TickMark }?
dchrt_ST_TickLblPos =
  string "high" | string "low" | string "nextTo" | string "none"
dchrt_CT_TickLblPos =

  ## default value: nextTo
  attribute val { dchrt_ST_TickLblPos }?
dchrt_ST_Skip = xsd:unsignedInt { minInclusive = "1" }
dchrt_CT_Skip = attribute val { dchrt_ST_Skip }
dchrt_ST_TimeUnit = string "days" | string "months" | string "years"
dchrt_CT_TimeUnit =

  ## default value: days
  attribute val { dchrt_ST_TimeUnit }?
dchrt_ST_AxisUnit = xsd:double { minExclusive = "0" }
```



```
dchrt_CT_AxisUnit = attribute val { dchrt_ST_AxisUnit }
dchrt_ST_BuiltInUnit =
  string "hundreds"
  | string "thousands"
  | string "tenThousands"
  | string "hundredThousands"
  | string "millions"
  | string "tenMillions"
  | string "hundredMillions"
  | string "billions"
  | string "trillions"
dchrt_CT_BuiltInUnit =

  ## default value: thousands
  attribute val { dchrt_ST_BuiltInUnit }?
dchrt_ST_PictureFormat =
  string "stretch" | string "stack" | string "stackScale"
dchrt_CT_PictureFormat = attribute val { dchrt_ST_PictureFormat }
dchrt_ST_PictureStackUnit = xsd:double { minExclusive = "0" }
dchrt_CT_PictureStackUnit = attribute val { dchrt_ST_PictureStackUnit }
dchrt_CT_PictureOptions =
  element applyToFront { dchrt_CT_Boolean }?,
  element applyToSides { dchrt_CT_Boolean }?,
  element applyToEnd { dchrt_CT_Boolean }?,
  element pictureFormat { dchrt_CT_PictureFormat }?,
  element pictureStackUnit { dchrt_CT_PictureStackUnit }?
dchrt_CT_DispUnitsLbl =
  element layout { dchrt_CT_Layout }?,
  element tx { dchrt_CT_Tx }?,
  element spPr { a_CT_ShapeProperties }?,
  element txPr { a_CT_TextBody }?
dchrt_CT_DispUnits =
  (element custUnit { dchrt_CT_Double }
   | element builtInUnit { dchrt_CT_BuiltInUnit }),
  element dispUnitsLbl { dchrt_CT_DispUnitsLbl }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_ST_Orientation = string "maxMin" | string "minMax"
dchrt_CT_Orientation =

  ## default value: minMax
  attribute val { dchrt_ST_Orientation }?
dchrt_ST_LogBase =
  xsd:double { minInclusive = "2" maxInclusive = "1000" }
dchrt_CT_LogBase = attribute val { dchrt_ST_LogBase }
dchrt_CT_Scaling =
  element logBase { dchrt_CT_LogBase }?,
  element orientation { dchrt_CT_Orientation }?,
  element max { dchrt_CT_Double }?,
  element min { dchrt_CT_Double }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_ST_LblOffset =
dchrt_ST_LblOffsetPercent | dchrt_ST_LblOffsetUShort
dchrt_ST_LblOffsetPercent =
```

```
  xsd:string {
    pattern = "0*(([0-9])|([1-9][0-9])|([1-9][0-9][0-9])|1000)%"
  }
dchrt_ST_LblOffsetUShort =
  xsd:unsignedShort { minInclusive = "0" maxInclusive = "1000" }
dchrt_CT_LblOffset =

  ## default value: 100%
  attribute val { dchrt_ST_LblOffset }?
dchrt_EG_AxShared =
  element axId { dchrt_CT_UnsignedInt },
  element scaling { dchrt_CT_Scaling },
  element delete { dchrt_CT_Boolean }?,
  element axPos { dchrt_CT_AxPos },
  element majorGridlines { dchrt_CT_ChartLines }?,
  element minorGridlines { dchrt_CT_ChartLines }?,
  element title { dchrt_CT_Title }?,
  element numFmt { dchrt_CT_NumFmt }?,
  element majorTickMark { dchrt_CT_TickMark }?,
  element minorTickMark { dchrt_CT_TickMark }?,
  element tickLblPos { dchrt_CT_TickLblPos }?,
  element spPr { a_CT_ShapeProperties }?,
  element txPr { a_CT_TextBody }?,
  element crossAx { dchrt_CT_UnsignedInt },
  (element crosses { dchrt_CT_Crosses }
   | element crossesAt { dchrt_CT_Double })?
dchrt_CT_CatAx =
  dchrt_EG_AxShared,
  element auto { dchrt_CT_Boolean }?,
  element lblAlgn { dchrt_CT_LblAlgn }?,
  element lblOffset { dchrt_CT_LblOffset }?,
  element tickLblSkip { dchrt_CT_Skip }?,
  element tickMarkSkip { dchrt_CT_Skip }?,
  element noMultiLvlLbl { dchrt_CT_Boolean }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_DateAx =
  dchrt_EG_AxShared,
  element auto { dchrt_CT_Boolean }?,
  element lblOffset { dchrt_CT_LblOffset }?,
  element baseTimeUnit { dchrt_CT_TimeUnit }?,
  element majorUnit { dchrt_CT_AxisUnit }?,
  element majorTimeUnit { dchrt_CT_TimeUnit }?,
  element minorUnit { dchrt_CT_AxisUnit }?,
  element minorTimeUnit { dchrt_CT_TimeUnit }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_SerAx =
  dchrt_EG_AxShared,
  element tickLblSkip { dchrt_CT_Skip }?,
  element tickMarkSkip { dchrt_CT_Skip }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_ValAx =
  dchrt_EG_AxShared,
  element crossBetween { dchrt_CT_CrossBetween }?,
```



```
  element majorUnit { dchrt_CT_AxisUnit }?,
  element minorUnit { dchrt_CT_AxisUnit }?,
  element dispUnits { dchrt_CT_DispUnits }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_PlotArea =
  element layout { dchrt_CT_Layout }?,
  (element areaChart { dchrt_CT_AreaChart }
   | element area3DChart { dchrt_CT_Area3DChart }
   | element lineChart { dchrt_CT_LineChart }
   | element line3DChart { dchrt_CT_Line3DChart }
   | element stockChart { dchrt_CT_StockChart }
   | element radarChart { dchrt_CT_RadarChart }
   | element scatterChart { dchrt_CT_ScatterChart }
   | element pieChart { dchrt_CT_PieChart }
   | element pie3DChart { dchrt_CT_Pie3DChart }
   | element doughnutChart { dchrt_CT_DoughnutChart }
   | element barChart { dchrt_CT_BarChart }
   | element bar3DChart { dchrt_CT_Bar3DChart }
   | element ofPieChart { dchrt_CT_OfPieChart }
   | element surfaceChart { dchrt_CT_SurfaceChart }
   | element surface3DChart { dchrt_CT_Surface3DChart }
   | element bubbleChart { dchrt_CT_BubbleChart })+,
  (element valAx { dchrt_CT_ValAx }
   | element catAx { dchrt_CT_CatAx }
   | element dateAx { dchrt_CT_DateAx }
   | element serAx { dchrt_CT_SerAx })*,
  element dTable { dchrt_CT_DTable }?,
  element spPr { a_CT_ShapeProperties }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_PivotFmt =
  element idx { dchrt_CT_UnsignedInt },
  element spPr { a_CT_ShapeProperties }?,
  element txPr { a_CT_TextBody }?,
  element marker { dchrt_CT_Marker }?,
  element dLbl { dchrt_CT_DLbl }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_PivotFmts = element pivotFmt { dchrt_CT_PivotFmt }*
dchrt_ST_LegendPos =
  string "b" | string "tr" | string "l" | string "r" | string "t"
dchrt_CT_LegendPos =

  ## default value: r
  attribute val { dchrt_ST_LegendPos }?
dchrt_EG_LegendEntryData = element txPr { a_CT_TextBody }?
dchrt_CT_LegendEntry =
  element idx { dchrt_CT_UnsignedInt },
  (element delete { dchrt_CT_Boolean }
   | dchrt_EG_LegendEntryData),
  element extLst { dchrt_CT_ExtensionList }?
dchrt_CT_Legend =
  element legendPos { dchrt_CT_LegendPos }?,
  element legendEntry { dchrt_CT_LegendEntry }*,
  element layout { dchrt_CT_Layout }?,
```

```
  element overlay { dchrt_CT_Boolean }?,
  element spPr { a_CT_ShapeProperties }?,
  element txPr { a_CT_TextBody }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_ST_DispBlanksAs = string "span" | string "gap" | string "zero"
dchrt_CT_DispBlanksAs =

  ## default value: zero
  attribute val { dchrt_ST_DispBlanksAs }?
dchrt_CT_Chart =
  element title { dchrt_CT_Title }?,
  element autoTitleDeleted { dchrt_CT_Boolean }?,
  element pivotFmts { dchrt_CT_PivotFmts }?,
  element view3D { dchrt_CT_View3D }?,
  element floor { dchrt_CT_Surface }?,
  element sideWall { dchrt_CT_Surface }?,
  element backWall { dchrt_CT_Surface }?,
  element plotArea { dchrt_CT_PlotArea },
  element legend { dchrt_CT_Legend }?,
  element plotVisOnly { dchrt_CT_Boolean }?,
  element dispBlanksAs { dchrt_CT_DispBlanksAs }?,
  element showDLblsOverMax { dchrt_CT_Boolean }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_ST_Style =
  xsd:unsignedByte { minInclusive = "1" maxInclusive = "48" }
dchrt_CT_Style = attribute val { dchrt_ST_Style }
dchrt_CT_PivotSource =
  element name { s_ST_Xstring },
  element fmtId { dchrt_CT_UnsignedInt },
  element extLst { dchrt_CT_ExtensionList }*
dchrt_CT_Protection =
  element chartObject { dchrt_CT_Boolean }?,
  element data { dchrt_CT_Boolean }?,
  element formatting { dchrt_CT_Boolean }?,
  element selection { dchrt_CT_Boolean }?,
  element userInterface { dchrt_CT_Boolean }?
dchrt_CT_HeaderFooter =

  ## default value: true
  attribute alignWithMargins { xsd:boolean }?,

  ## default value: false
  attribute differentOddEven { xsd:boolean }?,

  ## default value: false
  attribute differentFirst { xsd:boolean }?,
  element oddHeader { s_ST_Xstring }?,
  element oddFooter { s_ST_Xstring }?,
  element evenHeader { s_ST_Xstring }?,
  element evenFooter { s_ST_Xstring }?,
  element firstHeader { s_ST_Xstring }?,
  element firstFooter { s_ST_Xstring }?
dchrt_CT_PageMargins =
```



```
  attribute l { xsd:double },
  attribute r { xsd:double },
  attribute t { xsd:double },
  attribute b { xsd:double },
  attribute header { xsd:double },
  attribute footer { xsd:double }
dchrt_ST_PageSetupOrientation =
  string "default" | string "portrait" | string "landscape"
dchrt_CT_ExternalData =
  r_id,
  element autoUpdate { dchrt_CT_Boolean }?
dchrt_CT_PageSetup =

  ## default value: 1
  attribute paperSize { xsd:unsignedInt }?,
  attribute paperHeight { s_ST_PositiveUniversalMeasure }?,
  attribute paperWidth { s_ST_PositiveUniversalMeasure }?,

  ## default value: 1
  attribute firstPageNumber { xsd:unsignedInt }?,

  ## default value: default
  attribute orientation { dchrt_ST_PageSetupOrientation }?,

  ## default value: false
  attribute blackAndWhite { xsd:boolean }?,

  ## default value: false
  attribute draft { xsd:boolean }?,

  ## default value: false
  attribute useFirstPageNumber { xsd:boolean }?,

  ## default value: 600
  attribute horizontalDpi { xsd:int }?,

  ## default value: 600
  attribute verticalDpi { xsd:int }?,

  ## default value: 1
  attribute copies { xsd:unsignedInt }?
dchrt_CT_PrintSettings =
  element headerFooter { dchrt_CT_HeaderFooter }?,
  element pageMargins { dchrt_CT_PageMargins }?,
  element pageSetup { dchrt_CT_PageSetup }?,
  element legacyDrawingHT { dchrt_CT_RelId }?
dchrt_CT_ChartSpace =
  element date1904 { dchrt_CT_Boolean }?,
  element lang { dchrt_CT_TextLanguageID }?,
  element roundedCorners { dchrt_CT_Boolean }?,
  element style { dchrt_CT_Style }?,
  element clrMapOvr { a_CT_ColorMapping }?,
  element pivotSource { dchrt_CT_PivotSource }?,
```

```
  element protection { dchrt_CT_Protection }?,
  element chart { dchrt_CT_Chart },
  element spPr { a_CT_ShapeProperties }?,
  element txPr { a_CT_TextBody }?,
  element externalData { dchrt_CT_ExternalData }?,
  element printSettings { dchrt_CT_PrintSettings }?,
  element userShapes { dchrt_CT_RelId }?,
  element extLst { dchrt_CT_ExtensionList }?
dchrt_chartSpace = element chartSpace { dchrt_CT_ChartSpace }
dchrt_userShapes = element userShapes { cdr_CT_Drawing }
dchrt_chart = element chart { dchrt_CT_RelId }
```

B.5.1.1 Part Schemas

B.5.1.1.1 Chart Part

This schema is available in the file DrawingML_Chart.rnc.

```
include "dml-chart.rnc"
include "shared-relationshipReference.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-picture.rnc"
include "dml-chartDrawing.rnc"
start = dchrt_chartSpace
```

B.5.1.1.2 Chart Drawing Part

This schema is available in the file DrawingML_Chart_Drawing.rnc.

```
include "dml-chart.rnc"
include "shared-relationshipReference.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-picture.rnc"
include "dml-chartDrawing.rnc"
start = dchrt_userShapes
```

B.5.2 DrawingML - Chart Drawing

This schema is available in the file dml-chartDrawing.rnc.

```
default namespace =
  "http://schemas.openxmlformats.org/drawingml/2006/chartDrawing"
namespace a = "http://schemas.openxmlformats.org/drawingml/2006/main"
namespace cdr =
  "http://schemas.openxmlformats.org/drawingml/2006/chartDrawing"
namespace o = "urn:schemas-microsoft-com:office:office"
```



```
namespace v = "urn:schemas-microsoft-com:vml"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"

cdr_CT_ShapeNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvSpPr { a_CT_NonVisualDrawingShapeProps }
cdr_CT_Shape =
  attribute macro { xsd:string }?,
  attribute textlink { xsd:string }?,

  ## default value: true
  attribute fLocksText { xsd:boolean }?,

  ## default value: false
  attribute fPublished { xsd:boolean }?,
  element nvSpPr { cdr_CT_ShapeNonVisual },
  element spPr { a_CT_ShapeProperties },
  element style { a_CT_ShapeStyle }?,
  element txBody { a_CT_TextBody }?
cdr_CT_ConnectorNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvCxnSpPr { a_CT_NonVisualConnectorProperties }
cdr_CT_Connector =
  attribute macro { xsd:string }?,

  ## default value: false
  attribute fPublished { xsd:boolean }?,
  element nvCxnSpPr { cdr_CT_ConnectorNonVisual },
  element spPr { a_CT_ShapeProperties },
  element style { a_CT_ShapeStyle }?
cdr_CT_PictureNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvPicPr { a_CT_NonVisualPictureProperties }
cdr_CT_Picture =
  attribute macro { xsd:string }?,

  ## default value: false
  attribute fPublished { xsd:boolean }?,
  element nvPicPr { cdr_CT_PictureNonVisual },
  element blipFill { a_CT_BlipFillProperties },
  element spPr { a_CT_ShapeProperties },
  element style { a_CT_ShapeStyle }?
cdr_CT_GraphicFrameNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvGraphicFramePr { a_CT_NonVisualGraphicFrameProperties }
cdr_CT_GraphicFrame =
  attribute macro { xsd:string }?,

  ## default value: false
  attribute fPublished { xsd:boolean }?,
  element nvGraphicFramePr { cdr_CT_GraphicFrameNonVisual },
  element xfrm { a_CT_Transform2D },
```

```
  a_graphic
cdr_CT_GroupShapeNonVisual =
  element cNvPr { a_CT_NonVisualDrawingProps },
  element cNvGrpSpPr { a_CT_NonVisualGroupDrawingShapeProps }
cdr_CT_GroupShape =
  element nvGrpSpPr { cdr_CT_GroupShapeNonVisual },
  element grpSpPr { a_CT_GroupShapeProperties },
  (element sp { cdr_CT_Shape }
   | element grpSp { cdr_CT_GroupShape }
   | element graphicFrame { cdr_CT_GraphicFrame }
   | element cxnSp { cdr_CT_Connector }
   | element pic { cdr_CT_Picture })*
cdr_EG_ObjectChoices =
  element sp { cdr_CT_Shape }
  | element grpSp { cdr_CT_GroupShape }
  | element graphicFrame { cdr_CT_GraphicFrame }
  | element cxnSp { cdr_CT_Connector }
  | element pic { cdr_CT_Picture }
cdr_ST_MarkerCoordinate =
  xsd:double { minInclusive = "0.0" maxInclusive = "1.0" }
cdr_CT_Marker =
  element x { cdr_ST_MarkerCoordinate },
  element y { cdr_ST_MarkerCoordinate }
cdr_CT_RelSizeAnchor =
  element from { cdr_CT_Marker },
  element to { cdr_CT_Marker },
  cdr_EG_ObjectChoices
cdr_CT_AbsSizeAnchor =
  element from { cdr_CT_Marker },
  element ext { a_CT_PositiveSize2D },
  cdr_EG_ObjectChoices
cdr_EG_Anchor =
  element relSizeAnchor { cdr_CT_RelSizeAnchor }
  | element absSizeAnchor { cdr_CT_AbsSizeAnchor }
cdr_CT_Drawing = cdr_EG_Anchor*
```

B.5.3 DrawingML - Diagrams

This schema is available in the file dml-diagram.rnc.

```
default namespace =
  "http://schemas.openxmlformats.org/drawingml/2006/diagram"
namespace a = "http://schemas.openxmlformats.org/drawingml/2006/main"
namespace ddgrm =
  "http://schemas.openxmlformats.org/drawingml/2006/diagram"
namespace o = "urn:schemas-microsoft-com:office:office"
namespace r =
  "http://schemas.openxmlformats.org/officeDocument/2006/relationships"
namespace s =
  "http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
namespace v = "urn:schemas-microsoft-com:vml"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"
```



```

ddgrm_CT_CTName =
  attribute lang { xsd:string }?,
  attribute val { xsd:string }
ddgrm_CT_CTDescription =
  attribute lang { xsd:string }?,
  attribute val { xsd:string }
ddgrm_CT_CTCategory =
  attribute type { xsd:anyURI },
  attribute pri { xsd:unsignedInt }
ddgrm_CT_CTCategories = element cat { ddgrm_CT_CTCategory }*
ddgrm_ST_ClrAppMethod = "span" | "cycle" | "repeat"
ddgrm_ST_HueDir = "cw" | "ccw"
ddgrm_CT_Colors =

  ## default value: span
  attribute meth { ddgrm_ST_ClrAppMethod }?,

  ## default value: cw
  attribute hueDir { ddgrm_ST_HueDir }?,
  a_EG_ColorChoice*
ddgrm_CT_CTStyleLabel =
  attribute name { xsd:string },
  element fillClrLst { ddgrm_CT_Colors }?,
  element linClrLst { ddgrm_CT_Colors }?,
  element effectClrLst { ddgrm_CT_Colors }?,
  element txLinClrLst { ddgrm_CT_Colors }?,
  element txFillClrLst { ddgrm_CT_Colors }?,
  element txEffectClrLst { ddgrm_CT_Colors }?,
  element extLst { a_CT_OfficeArtExtensionList }?
ddgrm_CT_ColorTransform =
  attribute uniqueId { xsd:string }?,

  attribute minVer { xsd:string }?,
  element title { ddgrm_CT_CTName }*,
  element desc { ddgrm_CT_CTDescription }*,
  element catLst { ddgrm_CT_CTCategories }?,
  element styleLbl { ddgrm_CT_CTStyleLabel }*,
  element extLst { a_CT_OfficeArtExtensionList }?
ddgrm_colorsDef = element colorsDef { ddgrm_CT_ColorTransform }
ddgrm_CT_ColorTransformHeader =
  attribute uniqueId { xsd:string },

  attribute minVer { xsd:string }?,

  ## default value: 0
  attribute resId { xsd:int }?,
  element title { ddgrm_CT_CTName }+,
  element desc { ddgrm_CT_CTDescription }+,
  element catLst { ddgrm_CT_CTCategories }?,
  element extLst { a_CT_OfficeArtExtensionList }?
ddgrm_colorsDefHdr =
  element colorsDefHdr { ddgrm_CT_ColorTransformHeader }
```

```
ddgrm_CT_ColorTransformHeaderLst =
  element colorsDefHdr { ddgrm_CT_ColorTransformHeader }*
ddgrm_colorsDefHdrLst =
  element colorsDefHdrLst { ddgrm_CT_ColorTransformHeaderLst }
ddgrm_ST_PtType =
  "node" | "asst" | "doc" | "pres" | "parTrans" | "sibTrans"
ddgrm_CT_Pt =
  attribute modelId { ddgrm_ST_ModelId },

  ## default value: node
  attribute type { ddgrm_ST_PtType }?,

  ## default value: 0
  attribute cxnId { ddgrm_ST_ModelId }?,
  element prSet { ddgrm_CT_ElemPropSet }?,
  element spPr { a_CT_ShapeProperties }?,
  element t { a_CT_TextBody }?,
  element extLst { a_CT_OfficeArtExtensionList }?
ddgrm_CT_PtList = element pt { ddgrm_CT_Pt }*
ddgrm_ST_CxnType =
  "parOf" | "presOf" | "presParOf" | "unknownRelationship"
ddgrm_CT_Cxn =
  attribute modelId { ddgrm_ST_ModelId },

  ## default value: parOf
  attribute type { ddgrm_ST_CxnType }?,
  attribute srcId { ddgrm_ST_ModelId },
  attribute destId { ddgrm_ST_ModelId },
  attribute srcOrd { xsd:unsignedInt },
  attribute destOrd { xsd:unsignedInt },

  ## default value: 0
  attribute parTransId { ddgrm_ST_ModelId }?,

  ## default value: 0
  attribute sibTransId { ddgrm_ST_ModelId }?,
  attribute presId { xsd:string }?,
  element extLst { a_CT_OfficeArtExtensionList }?
ddgrm_CT_CxnList = element cxn { ddgrm_CT_Cxn }*
ddgrm_CT_DataModel =
  element ptLst { ddgrm_CT_PtList },
  element cxnLst { ddgrm_CT_CxnList }?,
  element bg { a_CT_BackgroundFormatting }?,
  element whole { a_CT_WholeE2oFormatting }?,
  element extLst { a_CT_OfficeArtExtensionList }?
ddgrm_dataModel = element dataModel { ddgrm_CT_DataModel }
ddgrm_AG_IteratorAttributes =

  ## default value: none
  attribute axis { ddgrm_ST_AxisTypes }?,

  ## default value: all
  attribute ptType { ddgrm_ST_ElementTypes }?,
```



```
  ## default value: true
  attribute hideLastTrans { ddgrm_ST_Booleans }?,

  ## default value: 1
  attribute st { ddgrm_ST_Ints }?,

  ## default value: 0
  attribute cnt { ddgrm_ST_UnsignedInts }?,

  ## default value: 1
  attribute step { ddgrm_ST_Ints }?
ddgrm_AG_ConstraintAttributes =
  attribute type { ddgrm_ST_ConstraintType },

  ## default value: self
  attribute for { ddgrm_ST_ConstraintRelationship }?,
  attribute forName { xsd:string }?,

  ## default value: all
  attribute ptType { ddgrm_ST_ElementType }?
ddgrm_AG_ConstraintRefAttributes =

  ## default value: none
  attribute refType { ddgrm_ST_ConstraintType }?,

  ## default value: self
  attribute refFor { ddgrm_ST_ConstraintRelationship }?,
  attribute refForName { xsd:string }?,

  ## default value: all
  attribute refPtType { ddgrm_ST_ElementType }?
ddgrm_CT_Constraint =
  ddgrm_AG_ConstraintAttributes,
  ddgrm_AG_ConstraintRefAttributes,

  ## default value: none
  attribute op { ddgrm_ST_BoolOperator }?,

  ## default value: 0
  attribute val { xsd:double }?,

  ## default value: 1
  attribute fact { xsd:double }?,
  element extLst { a_CT_OfficeArtExtensionList }?
ddgrm_CT_Constraints = element constr { ddgrm_CT_Constraint }*
ddgrm_CT_NumericRule =
  ddgrm_AG_ConstraintAttributes,

  ## default value: NaN
  attribute val { xsd:double }?,

  ## default value: NaN
```

```
  attribute fact { xsd:double }?,

  ## default value: NaN
  attribute max { xsd:double }?,
  element extLst { a_CT_OfficeArtExtensionList }?
ddgrm_CT_Rules = element rule { ddgrm_CT_NumericRule }*
ddgrm_CT_PresentationOf =
  ddgrm_AG_IteratorAttributes,
  element extLst { a_CT_OfficeArtExtensionList }?
ddgrm_ST_LayoutShapeType = a_ST_ShapeType | ddgrm_ST_OutputShapeType
ddgrm_ST_Index1 = xsd:unsignedInt { minInclusive = "1" }
ddgrm_CT_Adj =
  attribute idx { ddgrm_ST_Index1 },
  attribute val { xsd:double }
ddgrm_CT_AdjLst = element adj { ddgrm_CT_Adj }*
ddgrm_CT_Shape =

  ## default value: 0
  attribute rot { xsd:double }?,

  ## default value: none
  attribute type { ddgrm_ST_LayoutShapeType }?,
  r_blip?,

  ## default value: 0
  attribute zOrderOff { xsd:int }?,

  ## default value: false
  attribute hideGeom { xsd:boolean }?,

  ## default value: false
  attribute lkTxEntry { xsd:boolean }?,

  ## default value: false
  attribute blipPhldr { xsd:boolean }?,
  element adjLst { ddgrm_CT_AdjLst }?,
  element extLst { a_CT_OfficeArtExtensionList }?
ddgrm_CT_Parameter =
  attribute type { ddgrm_ST_ParameterId },
  attribute val { ddgrm_ST_ParameterVal }
ddgrm_CT_Algorithm =
  attribute type { ddgrm_ST_AlgorithmType },

  ## default value: 0
  attribute rev { xsd:unsignedInt }?,
  element param { ddgrm_CT_Parameter }*,
  element extLst { a_CT_OfficeArtExtensionList }?
ddgrm_CT_LayoutNode =
  attribute name { xsd:string }?,
  attribute styleLbl { xsd:string }?,

  ## default value: b
  attribute chOrder { ddgrm_ST_ChildOrderType }?,
```



```
  attribute moveWith { xsd:string }?,
  (element alg { ddgrm_CT_Algorithm }?
   | element shape { ddgrm_CT_Shape }?
   | element presOf { ddgrm_CT_PresentationOf }?
   | element constrLst { ddgrm_CT_Constraints }?
   | element ruleLst { ddgrm_CT_Rules }?
   | element varLst { ddgrm_CT_LayoutVariablePropertySet }?
   | element forEach { ddgrm_CT_ForEach }
   | element layoutNode { ddgrm_CT_LayoutNode }
   | element choose { ddgrm_CT_Choose }
   | element extLst { a_CT_OfficeArtExtensionList }?)*
ddgrm_CT_ForEach =
  attribute name { xsd:string }?,
  attribute ref { xsd:string }?,
  ddgrm_AG_IteratorAttributes,
  (element alg { ddgrm_CT_Algorithm }?
   | element shape { ddgrm_CT_Shape }?
   | element presOf { ddgrm_CT_PresentationOf }?
   | element constrLst { ddgrm_CT_Constraints }?
   | element ruleLst { ddgrm_CT_Rules }?
   | element forEach { ddgrm_CT_ForEach }
   | element layoutNode { ddgrm_CT_LayoutNode }
   | element choose { ddgrm_CT_Choose }
   | element extLst { a_CT_OfficeArtExtensionList }?)*
ddgrm_CT_When =
  attribute name { xsd:string }?,
  ddgrm_AG_IteratorAttributes,
  attribute func { ddgrm_ST_FunctionType },

  ## default value: none
  attribute arg { ddgrm_ST_FunctionArgument }?,
  attribute op { ddgrm_ST_FunctionOperator },
  attribute val { ddgrm_ST_FunctionValue },
  (element alg { ddgrm_CT_Algorithm }?
   | element shape { ddgrm_CT_Shape }?
   | element presOf { ddgrm_CT_PresentationOf }?
   | element constrLst { ddgrm_CT_Constraints }?
   | element ruleLst { ddgrm_CT_Rules }?
   | element forEach { ddgrm_CT_ForEach }
   | element layoutNode { ddgrm_CT_LayoutNode }
   | element choose { ddgrm_CT_Choose }
   | element extLst { a_CT_OfficeArtExtensionList }?)*
ddgrm_CT_Otherwise =
  attribute name { xsd:string }?,
  (element alg { ddgrm_CT_Algorithm }?
   | element shape { ddgrm_CT_Shape }?
   | element presOf { ddgrm_CT_PresentationOf }?
   | element constrLst { ddgrm_CT_Constraints }?
   | element ruleLst { ddgrm_CT_Rules }?
   | element forEach { ddgrm_CT_ForEach }
   | element layoutNode { ddgrm_CT_LayoutNode }
   | element choose { ddgrm_CT_Choose }
   | element extLst { a_CT_OfficeArtExtensionList }?)*
```

```
ddgrm_CT_Choose =
  attribute name { xsd:string }?,
  element if { ddgrm_CT_When }+,
  element else { ddgrm_CT_Otherwise }?
ddgrm_CT_SampleData =

  ## default value: false
  attribute useDef { xsd:boolean }?,
  element dataModel { ddgrm_CT_DataModel }?
ddgrm_CT_Category =
  attribute type { xsd:anyURI },
  attribute pri { xsd:unsignedInt }
ddgrm_CT_Categories = element cat { ddgrm_CT_Category }*
ddgrm_CT_Name =
  attribute lang { xsd:string }?,
  attribute val { xsd:string }
ddgrm_CT_Description =
  attribute lang { xsd:string }?,
  attribute val { xsd:string }
ddgrm_CT_DiagramDefinition =
  attribute uniqueId { xsd:string }?,

  attribute minVer { xsd:string }?,
  attribute defStyle { xsd:string }?,
  element title { ddgrm_CT_Name }*,
  element desc { ddgrm_CT_Description }*,
  element catLst { ddgrm_CT_Categories }?,
  element sampData { ddgrm_CT_SampleData }?,
  element styleData { ddgrm_CT_SampleData }?,
  element clrData { ddgrm_CT_SampleData }?,
  element layoutNode { ddgrm_CT_LayoutNode },
  element extLst { a_CT_OfficeArtExtensionList }?
ddgrm_layoutDef = element layoutDef { ddgrm_CT_DiagramDefinition }
ddgrm_CT_DiagramDefinitionHeader =
  attribute uniqueId { xsd:string },

  attribute minVer { xsd:string }?,
  attribute defStyle { xsd:string }?,

  ## default value: 0
  attribute resId { xsd:int }?,
  element title { ddgrm_CT_Name }+,
  element desc { ddgrm_CT_Description }+,
  element catLst { ddgrm_CT_Categories }?,
  element extLst { a_CT_OfficeArtExtensionList }?
ddgrm_layoutDefHdr =
  element layoutDefHdr { ddgrm_CT_DiagramDefinitionHeader }
ddgrm_CT_DiagramDefinitionHeaderLst =
  element layoutDefHdr { ddgrm_CT_DiagramDefinitionHeader }*
ddgrm_layoutDefHdrLst =
  element layoutDefHdrLst { ddgrm_CT_DiagramDefinitionHeaderLst }
ddgrm_CT_RelIds = r_dm, r_lo, r_qs, r_cs
ddgrm_relIds = element relIds { ddgrm_CT_RelIds }
```



```
ddgrm_ST_ParameterVal =
  ddgrm_ST_DiagramHorizontalAlignment
  | ddgrm_ST_VerticalAlignment
  | ddgrm_ST_ChildDirection
  | ddgrm_ST_ChildAlignment
  | ddgrm_ST_SecondaryChildAlignment
  | ddgrm_ST_LinearDirection
  | ddgrm_ST_SecondaryLinearDirection
  | ddgrm_ST_StartingElement
  | ddgrm_ST_BendPoint
  | ddgrm_ST_ConnectorRouting
  | ddgrm_ST_ArrowheadStyle
  | ddgrm_ST_ConnectorDimension
  | ddgrm_ST_RotationPath
  | ddgrm_ST_CenterShapeMapping
  | ddgrm_ST_NodeHorizontalAlignment
  | ddgrm_ST_NodeVerticalAlignment
  | ddgrm_ST_FallbackDimension
  | ddgrm_ST_TextDirection
  | ddgrm_ST_PyramidAccentPosition
  | ddgrm_ST_PyramidAccentTextMargin
  | ddgrm_ST_TextBlockDirection
  | ddgrm_ST_TextAnchorHorizontal
  | ddgrm_ST_TextAnchorVertical
  | ddgrm_ST_DiagramTextAlignment
  | ddgrm_ST_AutoTextRotation
  | ddgrm_ST_GrowDirection
  | ddgrm_ST_FlowDirection
  | ddgrm_ST_ContinueDirection
  | ddgrm_ST_Breakpoint
  | ddgrm_ST_Offset
  | ddgrm_ST_HierarchyAlignment
  | xsd:int
  | xsd:double
  | xsd:boolean
  | xsd:string
  | ddgrm_ST_ConnectorPoint
ddgrm_ST_ModelId = xsd:int | s_ST_Guid
ddgrm_ST_PrSetCustVal = s_ST_Percentage | xsd:int
ddgrm_CT_ElemPropSet =
  attribute presAssocID { ddgrm_ST_ModelId }?,
  attribute presName { xsd:string }?,
  attribute presStyleLbl { xsd:string }?,
  attribute presStyleIdx { xsd:int }?,
  attribute presStyleCnt { xsd:int }?,
  attribute loTypeId { xsd:string }?,
  attribute loCatId { xsd:string }?,
  attribute qsTypeId { xsd:string }?,
  attribute qsCatId { xsd:string }?,
  attribute csTypeId { xsd:string }?,
  attribute csCatId { xsd:string }?,
  attribute coherent3DOff { xsd:boolean }?,
  attribute phldrT { xsd:string }?,
```

```
  attribute phldr { xsd:boolean }?,
  attribute custAng { xsd:int }?,
  attribute custFlipVert { xsd:boolean }?,
  attribute custFlipHor { xsd:boolean }?,
  attribute custSzX { xsd:int }?,
  attribute custSzY { xsd:int }?,
  attribute custScaleX { ddgrm_ST_PrSetCustVal}?,
  attribute custScaleY { ddgrm_ST_PrSetCustVal}?,
  attribute custT { xsd:boolean }?,
  attribute custLinFactX { ddgrm_ST_PrSetCustVal}?,
  attribute custLinFactY { ddgrm_ST_PrSetCustVal}?,
  attribute custLinFactNeighborX { ddgrm_ST_PrSetCustVal}?,
  attribute custLinFactNeighborY { ddgrm_ST_PrSetCustVal}?,
  attribute custRadScaleRad { ddgrm_ST_PrSetCustVal}?,
  attribute custRadScaleInc { ddgrm_ST_PrSetCustVal}?,
  element presLayoutVars { ddgrm_CT_LayoutVariablePropertySet }?,
  element style { a_CT_ShapeStyle }?
ddgrm_ST_Direction = "norm" | "rev"
ddgrm_ST_HierBranchStyle = "l" | "r" | "hang" | "std" | "init"
ddgrm_ST_AnimOneStr = "none" | "one" | "branch"
ddgrm_ST_AnimLvlStr = "none" | "lvl" | "ctr"
ddgrm_CT_OrgChart =

  ## default value: false
  attribute val { xsd:boolean }?
ddgrm_ST_NodeCount = xsd:int { minInclusive = "-1" }
ddgrm_CT_ChildMax =

  ## default value: -1
  attribute val { ddgrm_ST_NodeCount }?
ddgrm_CT_ChildPref =

  ## default value: -1
  attribute val { ddgrm_ST_NodeCount }?
ddgrm_CT_BulletEnabled =

  ## default value: false
  attribute val { xsd:boolean }?
ddgrm_CT_Direction =

  ## default value: norm
  attribute val { ddgrm_ST_Direction }?
ddgrm_CT_HierBranchStyle =

  ## default value: std
  attribute val { ddgrm_ST_HierBranchStyle }?
ddgrm_CT_AnimOne =

  ## default value: one
  attribute val { ddgrm_ST_AnimOneStr }?
ddgrm_CT_AnimLvl =

  ## default value: none
```



```
  attribute val { ddgrm_ST_AnimLvlStr }?
ddgrm_ST_ResizeHandlesStr = "exact" | "rel"
ddgrm_CT_ResizeHandles =

  ## default value: rel
  attribute val { ddgrm_ST_ResizeHandlesStr }?
ddgrm_CT_LayoutVariablePropertySet =
  element orgChart { ddgrm_CT_OrgChart }?,
  element chMax { ddgrm_CT_ChildMax }?,
  element chPref { ddgrm_CT_ChildPref }?,
  element bulletEnabled { ddgrm_CT_BulletEnabled }?,
  element dir { ddgrm_CT_Direction }?,
  element hierBranch { ddgrm_CT_HierBranchStyle }?,
  element animOne { ddgrm_CT_AnimOne }?,
  element animLvl { ddgrm_CT_AnimLvl }?,
  element resizeHandles { ddgrm_CT_ResizeHandles }?
ddgrm_CT_SDName =
  attribute lang { xsd:string }?,
  attribute val { xsd:string }
ddgrm_CT_SDDescription =
  attribute lang { xsd:string }?,
  attribute val { xsd:string }
ddgrm_CT_SDCategory =
  attribute type { xsd:anyURI },
  attribute pri { xsd:unsignedInt }
ddgrm_CT_SDCategories = element cat { ddgrm_CT_SDCategory }*
ddgrm_CT_TextProps = a_EG_Text3D?
ddgrm_CT_StyleLabel =
  attribute name { xsd:string },
  element scene3d { a_CT_Scene3D }?,
  element sp3d { a_CT_Shape3D }?,
  element txPr { ddgrm_CT_TextProps }?,
  element style { a_CT_ShapeStyle }?,
  element extLst { a_CT_OfficeArtExtensionList }?
ddgrm_CT_StyleDefinition =
  attribute uniqueId { xsd:string }?,

  attribute minVer { xsd:string }?,
  element title { ddgrm_CT_SDName }*,
  element desc { ddgrm_CT_SDDescription }*,
  element catLst { ddgrm_CT_SDCategories }?,
  element scene3d { a_CT_Scene3D }?,
  element styleLbl { ddgrm_CT_StyleLabel }+,
  element extLst { a_CT_OfficeArtExtensionList }?
ddgrm_styleDef = element styleDef { ddgrm_CT_StyleDefinition }
ddgrm_CT_StyleDefinitionHeader =
  attribute uniqueId { xsd:string },

  attribute minVer { xsd:string }?,

  ## default value: 0
  attribute resId { xsd:int }?,
  element title { ddgrm_CT_SDName }+,
```

```
  element desc { ddgrm_CT_SDDescription }+,
  element catLst { ddgrm_CT_SDCategories }?,
  element extLst { a_CT_OfficeArtExtensionList }?
ddgrm_styleDefHdr =
  element styleDefHdr { ddgrm_CT_StyleDefinitionHeader }
ddgrm_CT_StyleDefinitionHeaderLst =
  element styleDefHdr { ddgrm_CT_StyleDefinitionHeader }*
ddgrm_styleDefHdrLst =
  element styleDefHdrLst { ddgrm_CT_StyleDefinitionHeaderLst }
ddgrm_ST_AlgorithmType =
  "composite"
  | "conn"
  | "cycle"
  | "hierChild"
  | "hierRoot"
  | "pyra"
  | "lin"
  | "sp"
  | "tx"
  | "snake"
ddgrm_ST_AxisType =
  "self"
  | "ch"
  | "des"
  | "desOrSelf"
  | "par"
  | "ancst"
  | "ancstOrSelf"
  | "followSib"
  | "precedSib"
  | "follow"
  | "preced"
  | "root"
  | "none"
ddgrm_ST_AxisTypes = list { ddgrm_ST_AxisType* }
ddgrm_ST_BoolOperator = "none" | "equ" | "gte" | "lte"
ddgrm_ST_ChildOrderType = "b" | "t"
ddgrm_ST_ConstraintType =
  "none"
  | "alignOff"
  | "begMarg"
  | "bendDist"
  | "begPad"
  | "b"
  | "bMarg"
  | "bOff"
  | "ctrX"
  | "ctrXOff"
  | "ctrY"
  | "ctrYOff"
  | "connDist"
  | "diam"
  | "endMarg"
```



```
  | "endPad"
  | "h"
  | "hArH"
  | "hOff"
  | "l"
  | "lMarg"
  | "lOff"
  | "r"
  | "rMarg"
  | "rOff"
  | "primFontSz"
  | "pyraAcctRatio"
  | "secFontSz"
  | "sibSp"
  | "secSibSp"
  | "sp"
  | "stemThick"
  | "t"
  | "tMarg"
  | "tOff"
  | "userA"
  | "userB"
  | "userC"
  | "userD"
  | "userE"
  | "userF"
  | "userG"
  | "userH"
  | "userI"
  | "userJ"
  | "userK"
  | "userL"
  | "userM"
  | "userN"
  | "userO"
  | "userP"
  | "userQ"
  | "userR"
  | "userS"
  | "userT"
  | "userU"
  | "userV"
  | "userW"
  | "userX"
  | "userY"
  | "userZ"
  | "w"
  | "wArH"
  | "wOff"
ddgrm_ST_ConstraintRelationship = "self" | "ch" | "des"
ddgrm_ST_ElementType =
  "all"
  | "doc"
```

```
  | "node"
  | "norm"
  | "nonNorm"
  | "asst"
  | "nonAsst"
  | "parTrans"
  | "pres"
  | "sibTrans"
ddgrm_ST_ElementTypes = list { ddgrm_ST_ElementType* }
ddgrm_ST_ParameterId =
  "horzAlign"
  | "vertAlign"
  | "chDir"
  | "chAlign"
  | "secChAlign"
  | "linDir"
  | "secLinDir"
  | "stElem"
  | "bendPt"
  | "connRout"
  | "begSty"
  | "endSty"
  | "dim"
  | "rotPath"
  | "ctrShpMap"
  | "nodeHorzAlign"
  | "nodeVertAlign"
  | "fallback"
  | "txDir"
  | "pyraAcctPos"
  | "pyraAcctTxMar"
  | "txBlDir"
  | "txAnchorHorz"
  | "txAnchorVert"
  | "txAnchorHorzCh"
  | "txAnchorVertCh"
  | "parTxLTRAlign"
  | "parTxRTLAlign"
  | "shpTxLTRAlignCh"
  | "shpTxRTLAlignCh"
  | "autoTxRot"
  | "grDir"
  | "flowDir"
  | "contDir"
  | "bkpt"
  | "off"
  | "hierAlign"
  | "bkPtFixedVal"
  | "stBulletLvl"
  | "stAng"
  | "spanAng"
  | "ar"
  | "lnSpPar"
```



```
  | "lnSpAfParP"
  | "lnSpCh"
  | "lnSpAfChP"
  | "rtShortDist"
  | "alignTx"
  | "pyraLvlNode"
  | "pyraAcctBkgdNode"
  | "pyraAcctTxNode"
  | "srcNode"
  | "dstNode"
  | "begPts"
  | "endPts"
ddgrm_ST_Ints = list { xsd:int* }
ddgrm_ST_UnsignedInts = list { xsd:unsignedInt* }
ddgrm_ST_Booleans = list { xsd:boolean* }
ddgrm_ST_FunctionType =
  "cnt"
  | "pos"
  | "revPos"
  | "posEven"
  | "posOdd"
  | "var"
  | "depth"
  | "maxDepth"
ddgrm_ST_FunctionOperator = "equ" | "neq" | "gt" | "lt" | "gte" | "lte"
ddgrm_ST_DiagramHorizontalAlignment = "l" | "ctr" | "r" | "none"
ddgrm_ST_VerticalAlignment = "t" | "mid" | "b" | "none"
ddgrm_ST_ChildDirection = "horz" | "vert"
ddgrm_ST_ChildAlignment = "t" | "b" | "l" | "r"
ddgrm_ST_SecondaryChildAlignment = "none" | "t" | "b" | "l" | "r"
ddgrm_ST_LinearDirection = "fromL" | "fromR" | "fromT" | "fromB"
ddgrm_ST_SecondaryLinearDirection =
  "none" | "fromL" | "fromR" | "fromT" | "fromB"
ddgrm_ST_StartingElement = "node" | "trans"
ddgrm_ST_RotationPath = "none" | "alongPath"
ddgrm_ST_CenterShapeMapping = "none" | "fNode"
ddgrm_ST_BendPoint = "beg" | "def" | "end"
ddgrm_ST_ConnectorRouting = "stra" | "bend" | "curve" | "longCurve"
ddgrm_ST_ArrowheadStyle = "auto" | "arr" | "noArr"
ddgrm_ST_ConnectorDimension = "1D" | "2D" | "cust"
ddgrm_ST_ConnectorPoint =
  "auto"
  | "bCtr"
  | "ctr"
  | "midL"
  | "midR"
  | "tCtr"
  | "bL"
  | "bR"
  | "tL"
  | "tR"
  | "radial"
ddgrm_ST_NodeHorizontalAlignment = "l" | "ctr" | "r"
```

```
ddgrm_ST_NodeVerticalAlignment = "t" | "mid" | "b"
ddgrm_ST_FallbackDimension = "1D" | "2D"
ddgrm_ST_TextDirection = "fromT" | "fromB"
ddgrm_ST_PyramidAccentPosition = "bef" | "aft"
ddgrm_ST_PyramidAccentTextMargin = "step" | "stack"
ddgrm_ST_TextBlockDirection = "horz" | "vert"
ddgrm_ST_TextAnchorHorizontal = "none" | "ctr"
ddgrm_ST_TextAnchorVertical = "t" | "mid" | "b"
ddgrm_ST_DiagramTextAlignment = "l" | "ctr" | "r"
ddgrm_ST_AutoTextRotation = "none" | "upr" | "grav"
ddgrm_ST_GrowDirection = "tL" | "tR" | "bL" | "bR"
ddgrm_ST_FlowDirection = "row" | "col"
ddgrm_ST_ContinueDirection = "revDir" | "sameDir"
ddgrm_ST_Breakpoint = "endCnv" | "bal" | "fixed"
ddgrm_ST_Offset = "ctr" | "off"
ddgrm_ST_HierarchyAlignment =
  "tL"
  | "tR"
  | "tCtrCh"
  | "tCtrDes"
  | "bL"
  | "bR"
  | "bCtrCh"
  | "bCtrDes"
  | "lT"
  | "lB"
  | "lCtrCh"
  | "lCtrDes"
  | "rT"
  | "rB"
  | "rCtrCh"
  | "rCtrDes"
ddgrm_ST_FunctionValue =
  xsd:int
  | xsd:boolean
  | ddgrm_ST_Direction
  | ddgrm_ST_HierBranchStyle
  | ddgrm_ST_AnimOneStr
  | ddgrm_ST_AnimLvlStr
  | ddgrm_ST_ResizeHandlesStr
ddgrm_ST_VariableType =
  "none"
  | "orgChart"
  | "chMax"
  | "chPref"
  | "bulEnabled"
  | "dir"
  | "hierBranch"
  | "animOne"
  | "animLvl"
  | "resizeHandles"
ddgrm_ST_FunctionArgument = ddgrm_ST_VariableType
ddgrm_ST_OutputShapeType = "none" | "conn"
```


B.5.3.1 Part Schemas

B.5.3.1.1 Diagram Colors Part

This schema is available in the file DrawingML_Diagram_Colors.rnc.

```
include "dml-diagram.rnc"
include "shared-relationshipReference.rnc"
include "dml-main.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = ddgrm_colorsDef
```

B.5.3.1.2 Diagram Data Part

This schema is available in the file DrawingML_Diagram_Data.rnc.

```
include "dml-diagram.rnc"
include "shared-relationshipReference.rnc"
include "dml-main.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = ddgrm_dataModel
```

B.5.3.1.3 Diagram Layout Definitions Part

This schema is available in the file DrawingML_Diagram_Layout_Definition.rnc.

```
include "dml-diagram.rnc"
include "shared-relationshipReference.rnc"
include "dml-main.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = ddgrm_layoutDef
```

B.5.3.1.4 Diagram Style Part

This schema is available in the file DrawingML_Diagram_Style.rnc.

```
include "dml-diagram.rnc"
include "shared-relationshipReference.rnc"
include "dml-main.rnc"
```

```
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
start = ddgrm_styleDef
```

B.6 VML

B.6.1 VML - Main

This schema is available in the file vml-main.rnc.

```
namespace o = "urn:schemas-microsoft-com:office:office"
namespace pvml = "urn:schemas-microsoft-com:office:powerpoint"
namespace r =
  "http://schemas.openxmlformats.org/officeDocument/2006/relationships"
namespace s =
  "http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
default namespace v = "urn:schemas-microsoft-com:vml"
namespace w =
  "http://schemas.openxmlformats.org/wordprocessingml/2006/main"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"

v_AG_Id = attribute id { xsd:string }?
v_AG_Style = attribute style { xsd:string }?
v_AG_Type = attribute type { xsd:string }?
v_AG_Adj = attribute adj { xsd:string }?
v_AG_Path = attribute path { xsd:string }?
v_AG_Fill =
  attribute filled { s_ST_TrueFalse }?,
  attribute fillcolor { s_ST_ColorType }?
v_AG_Chromakey = attribute chromakey { s_ST_ColorType }?
v_AG_Ext = attribute v:ext { v_ST_Ext }?
v_AG_CoreAttributes =
  v_AG_Id,
  v_AG_Style,
  attribute href { xsd:string }?,
  attribute target { xsd:string }?,
  attribute class { xsd:string }?,
  attribute title { xsd:string }?,
  attribute alt { xsd:string }?,
  attribute coordsize { xsd:string }?,
  attribute coordorigin { xsd:string }?,
  attribute wrapcoords { xsd:string }?,
  attribute print { s_ST_TrueFalse }?
v_AG_ShapeAttributes =
  v_AG_Chromakey,
  v_AG_Fill,
  attribute opacity { xsd:string }?,
  attribute stroked { s_ST_TrueFalse }?,
```



```
  attribute strokecolor { s_ST_ColorType }?,
  attribute strokeweight { xsd:string }?,
  attribute insetpen { s_ST_TrueFalse }?
v_AG_OfficeCoreAttributes =
  o_spid?,
  o_oned?,
  o_regroupid?,
  o_doubleclicknotify?,
  o_button?,
  o_userhidden?,
  o_bullet?,
  o_hr?,
  o_hrstd?,
  o_hrnoshade?,
  o_hrpct?,
  o_hralign?,
  o_allowincell?,
  o_allowoverlap?,
  o_userdrawn?,
  o_bordertopcolor?,
  o_borderleftcolor?,
  o_borderbottomcolor?,
  o_borderrightcolor?,
  o_dgmlayout?,
  o_dgmnodekind?,
  o_dgmlayoutmru?,
  o_insetmode?
v_AG_OfficeShapeAttributes =
  o_spt?,
  o_connectortype?,
  o_bwmode?,
  o_bwpure?,
  o_bwnormal?,
  o_forcedash?,
  o_oleicon?,
  o_ole?,
  o_preferrelative?,
  o_cliptowrap?,
  o_clip?
v_AG_AllCoreAttributes = v_AG_CoreAttributes, v_AG_OfficeCoreAttributes
v_AG_AllShapeAttributes =
  v_AG_ShapeAttributes, v_AG_OfficeShapeAttributes
v_AG_ImageAttributes =
  attribute src { xsd:string }?,
  attribute cropleft { xsd:string }?,
  attribute croptop { xsd:string }?,
  attribute cropright { xsd:string }?,
  attribute cropbottom { xsd:string }?,
  attribute gain { xsd:string }?,
  attribute blacklevel { xsd:string }?,
  attribute gamma { xsd:string }?,
  attribute grayscale { s_ST_TrueFalse }?,
  attribute bilevel { s_ST_TrueFalse }?
```

```
v_AG_StrokeAttributes =
  attribute on { s_ST_TrueFalse }?,
  attribute weight { xsd:string }?,
  attribute color { s_ST_ColorType }?,
  attribute opacity { xsd:string }?,
  attribute linestyle { v_ST_StrokeLineStyle }?,
  attribute miterlimit { xsd:decimal }?,
  attribute joinstyle { v_ST_StrokeJoinStyle }?,
  attribute endcap { v_ST_StrokeEndCap }?,
  attribute dashstyle { xsd:string }?,
  attribute filltype { v_ST_FillType }?,
  attribute src { xsd:string }?,
  attribute imageaspect { v_ST_ImageAspect }?,
  attribute imagesize { xsd:string }?,
  attribute imagealignshape { s_ST_TrueFalse }?,
  attribute color2 { s_ST_ColorType }?,
  attribute startarrow { v_ST_StrokeArrowType }?,
  attribute startarrowwidth { v_ST_StrokeArrowWidth }?,
  attribute startarrowlength { v_ST_StrokeArrowLength }?,
  attribute endarrow { v_ST_StrokeArrowType }?,
  attribute endarrowwidth { v_ST_StrokeArrowWidth }?,
  attribute endarrowlength { v_ST_StrokeArrowLength }?,
  o_href?,
  o_althref?,
  o_title?,
  o_forcedash?,
  r_id?,
  attribute insetpen { s_ST_TrueFalse }?,
  o_relid?
v_EG_ShapeElements =
  v_path
  | v_formulas
  | v_handles
  | v_fill
  | v_stroke
  | v_shadow
  | v_textbox
  | v_textpath
  | v_imagedata
  | o_skew
  | o_extrusion
  | o_callout
  | o_lock
  | o_clippath
  | o_signatureline
  | w10_wrap
  | w10_anchorlock
  | w10_bordertop
  | w10_borderbottom
  | w10_borderleft
  | w10_borderright
  | x_ClientData?
  | pvml_textdata?
```



```
v_shape = element shape { v_CT_Shape }
v_shapetype = element shapetype { v_CT_Shapetype }
v_group = element group { v_CT_Group }
v_background = element background { v_CT_Background }
v_CT_Shape =
  v_AG_AllCoreAttributes,
  v_AG_AllShapeAttributes,
  v_AG_Type,
  v_AG_Adj,
  v_AG_Path,
  o_gfxdata?,
  attribute equationxml { xsd:string }?,
  (v_EG_ShapeElements | o_ink | pvml_iscomment | o_equationxml)+
v_CT_Shapetype =
  v_AG_AllCoreAttributes,
  v_AG_AllShapeAttributes,
  v_AG_Adj,
  v_AG_Path,
  o_master?,
  v_EG_ShapeElements*,
  o_complex?
v_CT_Group =
  v_AG_AllCoreAttributes,
  v_AG_Fill,
  attribute editas { v_ST_EditAs }?,
  o_tableproperties?,
  o_tablelimits?,
  (v_EG_ShapeElements
   | v_group
   | v_shape
   | v_shapetype
   | v_arc
   | v_curve
   | v_image
   | v_line
   | v_oval
   | v_polyline
   | v_rect
   | v_roundrect
   | o_diagram)+
v_CT_Background =
  v_AG_Id,
  v_AG_Fill,
  o_bwmode?,
  o_bwpure?,
  o_bwnormal?,
  o_targetscreensize?,
  v_fill?
v_fill = element fill { v_CT_Fill }
v_formulas = element formulas { v_CT_Formulas }
v_handles = element handles { v_CT_Handles }
v_imagedata = element imagedata { v_CT_ImageData }
v_path = element path { v_CT_Path }
```

```
v_textbox = element textbox { v_CT_Textbox }
v_shadow = element shadow { v_CT_Shadow }
v_stroke = element stroke { v_CT_Stroke }
v_textpath = element textpath { v_CT_TextPath }
v_CT_Fill =
  v_AG_Id,
  attribute type { v_ST_FillType }?,
  attribute on { s_ST_TrueFalse }?,
  attribute color { s_ST_ColorType }?,
  attribute opacity { xsd:string }?,
  attribute color2 { s_ST_ColorType }?,
  attribute src { xsd:string }?,
  o_href?,
  o_althref?,
  attribute size { xsd:string }?,
  attribute origin { xsd:string }?,
  attribute position { xsd:string }?,
  attribute aspect { v_ST_ImageAspect }?,
  attribute colors { xsd:string }?,
  attribute angle { xsd:decimal }?,
  attribute alignshape { s_ST_TrueFalse }?,
  attribute focus { xsd:string }?,
  attribute focussize { xsd:string }?,
  attribute focusposition { xsd:string }?,
  attribute method { v_ST_FillMethod }?,
  o_detectmouseclick?,
  o_title?,
  o_opacity2?,
  attribute recolor { s_ST_TrueFalse }?,
  attribute rotate { s_ST_TrueFalse }?,
  r_id?,
  o_relid?,
  o_fill?
v_CT_Formulas = element f { v_CT_F }*
v_CT_F = attribute eqn { xsd:string }?
v_CT_Handles = element h { v_CT_H }*
v_CT_H =
  attribute position { xsd:string }?,
  attribute polar { xsd:string }?,
  attribute map { xsd:string }?,
  attribute invx { s_ST_TrueFalse }?,
  attribute invy { s_ST_TrueFalse }?,
  attribute switch { s_ST_TrueFalseBlank }?,
  attribute xrange { xsd:string }?,
  attribute yrange { xsd:string }?,
  attribute radiusrange { xsd:string }?
v_CT_ImageData =
  v_AG_Id,
  v_AG_ImageAttributes,
  v_AG_Chromakey,
  attribute embosscolor { s_ST_ColorType }?,
  attribute recolortarget { s_ST_ColorType }?,
  o_href?,
```



```
  o_althref?,
  o_title?,
  o_oleid?,
  o_detectmouseclick?,
  o_movie?,
  o_relid?,
  r_id?,
  r_pict?,
  r_href?
v_CT_Path =
  v_AG_Id,
  attribute v { xsd:string }?,
  attribute limo { xsd:string }?,
  attribute textboxrect { xsd:string }?,
  attribute fillok { s_ST_TrueFalse }?,
  attribute strokeok { s_ST_TrueFalse }?,
  attribute shadowok { s_ST_TrueFalse }?,
  attribute arrowok { s_ST_TrueFalse }?,
  attribute gradientshapeok { s_ST_TrueFalse }?,
  attribute textpathok { s_ST_TrueFalse }?,
  attribute insetpenok { s_ST_TrueFalse }?,
  o_connecttype?,
  o_connectlocs?,
  o_connectangles?,
  o_extrusionok?
v_CT_Shadow =
  v_AG_Id,
  attribute on { s_ST_TrueFalse }?,
  attribute type { v_ST_ShadowType }?,
  attribute obscured { s_ST_TrueFalse }?,
  attribute color { s_ST_ColorType }?,
  attribute opacity { xsd:string }?,
  attribute offset { xsd:string }?,
  attribute color2 { s_ST_ColorType }?,
  attribute offset2 { xsd:string }?,
  attribute origin { xsd:string }?,
  attribute matrix { xsd:string }?
v_CT_Stroke =
  v_AG_Id,
  v_AG_StrokeAttributes,
  o_left?,
  o_top?,
  o_right?,
  o_bottom?,
  o_column?
v_CT_Textbox =
  v_AG_Id,
  v_AG_Style,
  attribute inset { xsd:string }?,
  o_singleclick?,
  o_insetmode?,
  (w_txbxContent? | anyHTMLElementAsLocalElement)
```

```
anyHTMLElementAsLocalElement = element local:* { anyAttribute*, text?,
  anyHTMLElementAsLocalElement* }
v_CT_TextPath =
  v_AG_Id,
  v_AG_Style,
  attribute on { s_ST_TrueFalse }?,
  attribute fitshape { s_ST_TrueFalse }?,
  attribute fitpath { s_ST_TrueFalse }?,
  attribute trim { s_ST_TrueFalse }?,
  attribute xscale { s_ST_TrueFalse }?,
  attribute string { xsd:string }?
v_arc = element arc { v_CT_Arc }
v_curve = element curve { v_CT_Curve }
v_image = element image { v_CT_Image }
v_line = element line { v_CT_Line }
v_oval = element oval { v_CT_Oval }
v_polyline = element polyline { v_CT_PolyLine }
v_rect = element rect { v_CT_Rect }
v_roundrect = element roundrect { v_CT_RoundRect }
v_CT_Arc =
  v_AG_AllCoreAttributes,
  v_AG_AllShapeAttributes,
  attribute startAngle { xsd:decimal }?,
  attribute endAngle { xsd:decimal }?,
  v_EG_ShapeElements*
v_CT_Curve =
  v_AG_AllCoreAttributes,
  v_AG_AllShapeAttributes,
  attribute from { xsd:string }?,
  attribute control1 { xsd:string }?,
  attribute control2 { xsd:string }?,
  attribute to { xsd:string }?,
  v_EG_ShapeElements*
v_CT_Image =
  v_AG_AllCoreAttributes,
  v_AG_AllShapeAttributes,
  v_AG_ImageAttributes,
  v_EG_ShapeElements*
v_CT_Line =
  v_AG_AllCoreAttributes,
  v_AG_AllShapeAttributes,
  attribute from { xsd:string }?,
  attribute to { xsd:string }?,
  v_EG_ShapeElements*
v_CT_Oval =
  v_AG_AllCoreAttributes,
  v_AG_AllShapeAttributes,
  (v_EG_ShapeElements*)+
v_CT_PolyLine =
  v_AG_AllCoreAttributes,
  v_AG_AllShapeAttributes,
  attribute points { xsd:string }?,
  (v_EG_ShapeElements | o_ink)*
```



```
v_CT_Rect =
  v_AG_AllCoreAttributes,
  v_AG_AllShapeAttributes,
  (v_EG_ShapeElements*)+
v_CT_RoundRect =
  v_AG_AllCoreAttributes,
  v_AG_AllShapeAttributes,
  attribute arcsize { xsd:string }?,
  (v_EG_ShapeElements*)+
v_ST_Ext = string "view" | string "edit" | string "backwardCompatible"
v_ST_FillType =
  string "solid"
  | string "gradient"
  | string "gradientRadial"
  | string "tile"
  | string "pattern"
  | string "frame"
v_ST_FillMethod =
  string "none"
  | string "linear"
  | string "sigma"
  | string "any"
  | string "linear sigma"
v_ST_ShadowType =
  string "single"
  | string "double"
  | string "emboss"
  | string "perspective"
v_ST_StrokeLineStyle =
  string "single"
  | string "thinThin"
  | string "thinThick"
  | string "thickThin"
  | string "thickBetweenThin"
v_ST_StrokeJoinStyle = string "round" | string "bevel" | string "miter"
v_ST_StrokeEndCap = string "flat" | string "square" | string "round"
v_ST_StrokeArrowLength =
  string "short" | string "medium" | string "long"
v_ST_StrokeArrowWidth =
  string "narrow" | string "medium" | string "wide"
v_ST_StrokeArrowType =
  string "none"
  | string "block"
  | string "classic"
  | string "oval"
  | string "diamond"
  | string "open"
v_ST_ImageAspect = string "ignore" | string "atMost" | string "atLeast"
v_ST_EditAs =
  string "canvas"
  | string "orgchart"
  | string "radial"
  | string "cycle"
```

```
  | string "stacked"
  | string "venn"
  | string "bullseye"
```

B.6.2 VML - Office Drawing

This schema is available in the file vml-officeDrawing.rnc.

```
default namespace o = "urn:schemas-microsoft-com:office:office"
namespace r =
  "http://schemas.openxmlformats.org/officeDocument/2006/relationships"
namespace s =
  "http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
namespace v = "urn:schemas-microsoft-com:vml"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"

o_bwmode = attribute o:bwmode { o_ST_BWMode }
o_bwpure = attribute o:bwpure { o_ST_BWMode }
o_bwnormal = attribute o:bwnormal { o_ST_BWMode }
o_targetscreensize = attribute o:targetscreensize { o_ST_ScreenSize }
o_insetmode =

  ## default value: custom
  attribute o:insetmode { o_ST_InsetMode }
o_spt = attribute o:spt { xsd:float }
o_wrapcoords = attribute o:wrapcoords { xsd:string }
o_oned = attribute o:oned { s_ST_TrueFalse }
o_regroupid = attribute o:regroupid { xsd:integer }
o_doubleclicknotify = attribute o:doubleclicknotify { s_ST_TrueFalse }
o_connectortype =

  ## default value: straight
  attribute o:connectortype { o_ST_ConnectorType }
o_button = attribute o:button { s_ST_TrueFalse }
o_userhidden = attribute o:userhidden { s_ST_TrueFalse }
o_forcedash = attribute o:forcedash { s_ST_TrueFalse }
o_oleicon = attribute o:oleicon { s_ST_TrueFalse }
o_ole = attribute o:ole { s_ST_TrueFalseBlank }
o_preferrelative = attribute o:preferrelative { s_ST_TrueFalse }
o_cliptowrap = attribute o:cliptowrap { s_ST_TrueFalse }
o_clip = attribute o:clip { s_ST_TrueFalse }
o_bullet = attribute o:bullet { s_ST_TrueFalse }
o_hr = attribute o:hr { s_ST_TrueFalse }
o_hrstd = attribute o:hrstd { s_ST_TrueFalse }
o_hrnoshade = attribute o:hrnoshade { s_ST_TrueFalse }
o_hrpct = attribute o:hrpct { xsd:float }
o_hralign =

  ## default value: left
  attribute o:hralign { o_ST_HrAlign }
o_allowincell = attribute o:allowincell { s_ST_TrueFalse }
o_allowoverlap = attribute o:allowoverlap { s_ST_TrueFalse }
```



```
o_userdrawn = attribute o:userdrawn { s_ST_TrueFalse }
o_bordertopcolor = attribute o:bordertopcolor { xsd:string }
o_borderleftcolor = attribute o:borderleftcolor { xsd:string }
o_borderbottomcolor = attribute o:borderbottomcolor { xsd:string }
o_borderrightcolor = attribute o:borderrightcolor { xsd:string }
o_connecttype = attribute o:connecttype { o_ST_ConnectType }
o_connectlocs = attribute o:connectlocs { xsd:string }
o_connectangles = attribute o:connectangles { xsd:string }
o_master = attribute o:master { xsd:string }
o_extrusionok = attribute o:extrusionok { s_ST_TrueFalse }
o_href = attribute o:href { xsd:string }
o_althref = attribute o:althref { xsd:string }
o_title = attribute o:title { xsd:string }
o_singleclick = attribute o:singleclick { s_ST_TrueFalse }
o_oleid = attribute o:oleid { xsd:float }
o_detectmouseclick = attribute o:detectmouseclick { s_ST_TrueFalse }
o_movie = attribute o:movie { xsd:float }
o_spid = attribute o:spid { xsd:string }
o_opacity2 = attribute o:opacity2 { xsd:string }
o_relid = attribute o:relid { r_ST_RelationshipId }
o_dgmlayout = attribute o:dgmlayout { o_ST_DiagramLayout }
o_dgmnodekind = attribute o:dgmnodekind { xsd:integer }
o_dgmlayoutmru = attribute o:dgmlayoutmru { o_ST_DiagramLayout }
o_gfxdata = attribute o:gfxdata { xsd:base64Binary }
o_tableproperties = attribute o:tableproperties { xsd:string }
o_tablelimits = attribute o:tablelimits { xsd:string }
o_shapedefaults = element shapedefaults { o_CT_ShapeDefaults }
o_shapelayout = element shapelayout { o_CT_ShapeLayout }
o_signatureline = element signatureline { o_CT_SignatureLine }
o_ink = element ink { o_CT_Ink }
o_diagram = element diagram { o_CT_Diagram }
o_equationxml = element equationxml { o_CT_EquationXml }
o_CT_ShapeDefaults =
  v_AG_Ext,
  attribute spidmax { xsd:integer }?,
  attribute style { xsd:string }?,
  attribute fill { s_ST_TrueFalse }?,
  attribute fillcolor { s_ST_ColorType }?,
  attribute stroke { s_ST_TrueFalse }?,
  attribute strokecolor { s_ST_ColorType }?,
  attribute o:allowincell { s_ST_TrueFalse }?,
  (v_fill?
   & v_stroke?
   & v_textbox?
   & v_shadow?
   & o_skew?
   & o_extrusion?
   & o_callout?
   & o_lock?
   & element colormru { o_CT_ColorMru }?
   & element colormenu { o_CT_ColorMenu }?)?
o_CT_Ink =
  attribute i { xsd:string }?,
```

```
  attribute annotation { s_ST_TrueFalse }?,
  attribute contentType { o_ST_ContentType }?,
  empty
o_CT_SignatureLine =
  v_AG_Ext,
  attribute issignatureline { s_ST_TrueFalse }?,
  attribute id { s_ST_Guid }?,
  attribute provid { s_ST_Guid }?,
  attribute signinginstructionsset { s_ST_TrueFalse }?,
  attribute allowcomments { s_ST_TrueFalse }?,
  attribute showsigndate { s_ST_TrueFalse }?,
  attribute o:suggestedsigner { xsd:string }?,
  attribute o:suggestedsigner2 { xsd:string }?,
  attribute o:suggestedsigneremail { xsd:string }?,
  attribute signinginstructions { xsd:string }?,
  attribute addlxml { xsd:string }?,
  attribute sigprovurl { xsd:string }?
o_CT_ShapeLayout =
  v_AG_Ext,
  (element idmap { o_CT_IdMap }?
   & element regrouptable { o_CT_RegroupTable }?
   & element rules { o_CT_Rules }?)
o_CT_IdMap =
  v_AG_Ext,
  attribute data { xsd:string }?
o_CT_RegroupTable =
  v_AG_Ext,
  element entry { o_CT_Entry }*
o_CT_Entry =
  attribute new { xsd:int }?,
  attribute old { xsd:int }?
o_CT_Rules =
  v_AG_Ext,
  element r { o_CT_R }*
o_CT_R =
  attribute id { xsd:string },
  attribute type { o_ST_RType }?,
  attribute how { o_ST_How }?,
  attribute idref { xsd:string }?,
  element proxy { o_CT_Proxy }*
o_CT_Proxy =

  ## default value: false
  attribute start { s_ST_TrueFalseBlank }?,

  ## default value: false
  attribute end { s_ST_TrueFalseBlank }?,
  attribute idref { xsd:string }?,
  attribute connectloc { xsd:int }?
o_CT_Diagram =
  v_AG_Ext,
  attribute dgmstyle { xsd:integer }?,
  attribute autoformat { s_ST_TrueFalse }?,
```



```
  attribute reverse { s_ST_TrueFalse }?,
  attribute autolayout { s_ST_TrueFalse }?,
  attribute dgmscalex { xsd:integer }?,
  attribute dgmscaley { xsd:integer }?,
  attribute dgmfontsize { xsd:integer }?,
  attribute constrainbounds { xsd:string }?,
  attribute dgmbasetextscale { xsd:integer }?,
  element relationtable { o_CT_RelationTable }?
o_CT_EquationXml =
  attribute contentType { o_ST_AlternateMathContentType }?,
  o_CT_EquationXml_any
o_CT_EquationXml_any =
  element * - (o:* | v:* | w10:* | x:*) {
    anyAttribute*,
    mixed { anyElement* }
  }
o_ST_AlternateMathContentType = xsd:string
o_CT_RelationTable =
  v_AG_Ext,
  element rel { o_CT_Relation }*
o_CT_Relation =
  v_AG_Ext,
  attribute idsrc { xsd:string }?,
  attribute iddest { xsd:string }?,
  attribute idcntr { xsd:string }?
o_CT_ColorMru =
  v_AG_Ext,
  attribute colors { xsd:string }?
o_CT_ColorMenu =
  v_AG_Ext,
  attribute strokecolor { s_ST_ColorType }?,
  attribute fillcolor { s_ST_ColorType }?,
  attribute shadowcolor { s_ST_ColorType }?,
  attribute extrusioncolor { s_ST_ColorType }?
o_skew = element skew { o_CT_Skew }
o_extrusion = element extrusion { o_CT_Extrusion }
o_callout = element callout { o_CT_Callout }
o_lock = element lock { o_CT_Lock }
o_OLEObject = element OLEObject { o_CT_OLEObject }
o_complex = element complex { o_CT_Complex }
o_left = element left { o_CT_StrokeChild }
o_top = element top { o_CT_StrokeChild }
o_right = element right { o_CT_StrokeChild }
o_bottom = element bottom { o_CT_StrokeChild }
o_column = element column { o_CT_StrokeChild }
o_clippath = element clippath { o_CT_ClipPath }
o_fill = element fill { o_CT_Fill }
o_CT_Skew =
  v_AG_Ext,
  attribute id { xsd:string }?,
  attribute on { s_ST_TrueFalse }?,
  attribute offset { xsd:string }?,
  attribute origin { xsd:string }?,
```

```
  attribute matrix { xsd:string }?
o_CT_Extrusion =
  v_AG_Ext,
  attribute on { s_ST_TrueFalse }?,

  ## default value: parallel
  attribute type { o_ST_ExtrusionType }?,

  ## default value: solid
  attribute render { o_ST_ExtrusionRender }?,
  attribute viewpointorigin { xsd:string }?,
  attribute viewpoint { xsd:string }?,

  ## default value: XY
  attribute plane { o_ST_ExtrusionPlane }?,
  attribute skewangle { xsd:float }?,
  attribute skewamt { xsd:string }?,
  attribute foredepth { xsd:string }?,
  attribute backdepth { xsd:string }?,
  attribute orientation { xsd:string }?,
  attribute orientationangle { xsd:float }?,
  attribute lockrotationcenter { s_ST_TrueFalse }?,
  attribute autorotationcenter { s_ST_TrueFalse }?,
  attribute rotationcenter { xsd:string }?,
  attribute rotationangle { xsd:string }?,
  attribute colormode { o_ST_ColorMode }?,
  attribute color { s_ST_ColorType }?,
  attribute shininess { xsd:float }?,
  attribute specularity { xsd:string }?,
  attribute diffusity { xsd:string }?,
  attribute metal { s_ST_TrueFalse }?,
  attribute edge { xsd:string }?,
  attribute facet { xsd:string }?,
  attribute lightface { s_ST_TrueFalse }?,
  attribute brightness { xsd:string }?,
  attribute lightposition { xsd:string }?,
  attribute lightlevel { xsd:string }?,
  attribute lightharsh { s_ST_TrueFalse }?,
  attribute lightposition2 { xsd:string }?,
  attribute lightlevel2 { xsd:string }?,
  attribute lightharsh2 { s_ST_TrueFalse }?
o_CT_Callout =
  v_AG_Ext,
  attribute on { s_ST_TrueFalse }?,
  attribute type { xsd:string }?,
  attribute gap { xsd:string }?,
  attribute angle { o_ST_Angle }?,
  attribute dropauto { s_ST_TrueFalse }?,
  attribute drop { o_ST_CalloutDrop }?,
  attribute distance { xsd:string }?,

  ## default value: f
  attribute lengthspecified { s_ST_TrueFalse }?,
```



```
  attribute length { xsd:string }?,
  attribute accentbar { s_ST_TrueFalse }?,
  attribute textborder { s_ST_TrueFalse }?,
  attribute minusx { s_ST_TrueFalse }?,
  attribute minusy { s_ST_TrueFalse }?
o_CT_Lock =
  v_AG_Ext,
  attribute position { s_ST_TrueFalse }?,
  attribute selection { s_ST_TrueFalse }?,
  attribute grouping { s_ST_TrueFalse }?,
  attribute ungrouping { s_ST_TrueFalse }?,
  attribute rotation { s_ST_TrueFalse }?,
  attribute cropping { s_ST_TrueFalse }?,
  attribute verticies { s_ST_TrueFalse }?,
  attribute adjusthandles { s_ST_TrueFalse }?,
  attribute text { s_ST_TrueFalse }?,
  attribute aspectratio { s_ST_TrueFalse }?,
  attribute shapetype { s_ST_TrueFalse }?
o_CT_OLEObject =
  attribute Type { o_ST_OLEType }?,
  attribute ProgID { xsd:string }?,
  attribute ShapeID { xsd:string }?,
  attribute DrawAspect { o_ST_OLEDrawAspect }?,
  attribute ObjectID { xsd:string }?,
  r_id?,
  attribute UpdateMode { o_ST_OLEUpdateMode }?,
  element LinkType { o_ST_OLELinkType }?,
  element LockedField { s_ST_TrueFalseBlank }?,
  element FieldCodes { xsd:string }?
o_CT_Complex = v_AG_Ext
o_CT_StrokeChild =
  v_AG_Ext,
  attribute on { s_ST_TrueFalse }?,
  attribute weight { xsd:string }?,
  attribute color { s_ST_ColorType }?,
  attribute color2 { s_ST_ColorType }?,
  attribute opacity { xsd:string }?,
  attribute linestyle { v_ST_StrokeLineStyle }?,
  attribute miterlimit { xsd:decimal }?,
  attribute joinstyle { v_ST_StrokeJoinStyle }?,
  attribute endcap { v_ST_StrokeEndCap }?,
  attribute dashstyle { xsd:string }?,
  attribute insetpen { s_ST_TrueFalse }?,
  attribute filltype { v_ST_FillType }?,
  attribute src { xsd:string }?,
  attribute imageaspect { v_ST_ImageAspect }?,
  attribute imagesize { xsd:string }?,
  attribute imagealignshape { s_ST_TrueFalse }?,
  attribute startarrow { v_ST_StrokeArrowType }?,
  attribute startarrowwidth { v_ST_StrokeArrowWidth }?,
  attribute startarrowlength { v_ST_StrokeArrowLength }?,
  attribute endarrow { v_ST_StrokeArrowType }?,
  attribute endarrowwidth { v_ST_StrokeArrowWidth }?,
```

```
  attribute endarrowlength { v_ST_StrokeArrowLength }?,
  o_href?,
  o_althref?,
  o_title?,
  o_forcedash?
o_CT_ClipPath = attribute o:v { xsd:string }
o_CT_Fill =
  v_AG_Ext,
  attribute type { o_ST_FillType }?
o_ST_RType =
  string "arc" | string "callout" | string "connector" | string "align"
o_ST_How =
  string "top"
  | string "middle"
  | string "bottom"
  | string "left"
  | string "center"
  | string "right"
o_ST_BWMode =
  string "color"
  | string "auto"
  | string "grayScale"
  | string "lightGrayscale"
  | string "inverseGray"
  | string "grayOutline"
  | string "highContrast"
  | string "black"
  | string "white"
  | string "hide"
  | string "undrawn"
  | string "blackTextAndLines"
o_ST_ScreenSize =
  string "544,376"
  | string "640,480"
  | string "720,512"
  | string "800,600"
  | string "1024,768"
  | string "1152,862"
o_ST_InsetMode = string "auto" | string "custom"
o_ST_ColorMode = string "auto" | string "custom"
o_ST_ContentType = xsd:string
o_ST_DiagramLayout = "0" | "1" | "2" | "3"
o_ST_ExtrusionType = string "perspective" | string "parallel"
o_ST_ExtrusionRender =
  string "solid" | string "wireFrame" | string "boundingCube"
o_ST_ExtrusionPlane = string "XY" | string "ZX" | string "YZ"
o_ST_Angle =
  string "any"
  | string "30"
  | string "45"
  | string "60"
  | string "90"
  | string "auto"
```



```
o_ST_CalloutDrop = xsd:string
o_ST_CalloutPlacement =
  string "top" | string "center" | string "bottom" | string "user"
o_ST_ConnectorType =
  string "none" | string "straight" | string "elbow" | string "curved"
o_ST_HrAlign = string "left" | string "right" | string "center"
o_ST_ConnectType =
  string "none" | string "rect" | string "segments" | string "custom"
o_ST_OLELinkType = xsd:string
o_ST_OLEType = string "Embed" | string "Link"
o_ST_OLEDrawAspect = string "Content" | string "Icon"
o_ST_OLEUpdateMode = string "Always" | string "OnCall"
o_ST_FillType =
  string "gradientCenter"
  | string "solid"
  | string "pattern"
  | string "tile"
  | string "frame"
  | string "gradientUnscaled"
  | string "gradientRadial"
  | string "gradient"
  | string "background"
o_any_vml_vml =
  v_shape
  | v_shapetype
  | v_group
  | v_background
  | v_fill
  | v_formulas
  | v_handles
  | v_imagedata
  | v_path
  | v_textbox
  | v_shadow
  | v_stroke
  | v_textpath
  | v_arc
  | v_curve
  | v_image
  | v_line
  | v_oval
  | v_polyline
  | v_rect
  | v_roundrect
```

B.6.3 VML - Wordprocessing Drawing

This schema is available in the file vml-wordprocessingDrawing.rnc.

```
default namespace = "urn:schemas-microsoft-com:office:word"
namespace o = "urn:schemas-microsoft-com:office:office"
namespace v = "urn:schemas-microsoft-com:vml"
namespace w10 = "urn:schemas-microsoft-com:office:word"
```



```
namespace x = "urn:schemas-microsoft-com:office:excel"

w10_bordertop = element bordertop { w10_CT_Border }
w10_borderleft = element borderleft { w10_CT_Border }
w10_borderright = element borderright { w10_CT_Border }
w10_borderbottom = element borderbottom { w10_CT_Border }
w10_CT_Border =
  attribute type { w10_ST_BorderType }?,
  attribute width { xsd:positiveInteger }?,
  attribute shadow { w10_ST_BorderShadow }?
w10_wrap = element wrap { w10_CT_Wrap }
w10_CT_Wrap =
  attribute type { w10_ST_WrapType }?,
  attribute side { w10_ST_WrapSide }?,
  attribute anchorx { w10_ST_HorizontalAnchor }?,
  attribute anchory { w10_ST_VerticalAnchor }?
w10_anchorlock = element anchorlock { w10_CT_AnchorLock }
w10_CT_AnchorLock = empty
w10_ST_BorderType =
  string "none"
  | string "single"
  | string "thick"
  | string "double"
  | string "hairline"
  | string "dot"
  | string "dash"
  | string "dotDash"
  | string "dashDotDot"
  | string "triple"
  | string "thinThickSmall"
  | string "thickThinSmall"
  | string "thickBetweenThinSmall"
  | string "thinThick"
  | string "thickThin"
  | string "thickBetweenThin"
  | string "thinThickLarge"
  | string "thickThinLarge"
  | string "thickBetweenThinLarge"
  | string "wave"
  | string "doubleWave"
  | string "dashedSmall"
  | string "dashDotStroked"
  | string "threeDEmboss"
  | string "threeDEngrave"
  | string "HTMLOutset"
  | string "HTMLInset"
w10_ST_BorderShadow =
  string "t" | string "true" | string "f" | string "false"
w10_ST_WrapType =
  string "topAndBottom"
  | string "square"
  | string "none"
  | string "tight"
```



```
  | string "through"
w10_ST_WrapSide =
  string "both" | string "left" | string "right" | string "largest"
w10_ST_HorizontalAnchor =
  string "margin" | string "page" | string "text" | string "char"
w10_ST_VerticalAnchor =
  string "margin" | string "page" | string "text" | string "line"
```

B.6.4 VML - Spreadsheet Drawing

This schema is available in the file vml-spreadsheetDrawing.rnc.

```
default namespace = "urn:schemas-microsoft-com:office:excel"
namespace o = "urn:schemas-microsoft-com:office:office"
namespace s =
  "http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
namespace v = "urn:schemas-microsoft-com:vml"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"

x_ClientData = element ClientData { x_CT_ClientData }
x_CT_ClientData =
  attribute ObjectType { x_ST_ObjectType },
  (element MoveWithCells { s_ST_TrueFalseBlank }
   | element SizeWithCells { s_ST_TrueFalseBlank }
   | element Anchor { xsd:string }
   | element Locked { s_ST_TrueFalseBlank }
   | element DefaultSize { s_ST_TrueFalseBlank }
   | element PrintObject { s_ST_TrueFalseBlank }
   | element Disabled { s_ST_TrueFalseBlank }
   | element AutoFill { s_ST_TrueFalseBlank }
   | element AutoLine { s_ST_TrueFalseBlank }
   | element AutoPict { s_ST_TrueFalseBlank }
   | element FmlaMacro { xsd:string }
   | element TextHAlign { xsd:string }
   | element TextVAlign { xsd:string }
   | element LockText { s_ST_TrueFalseBlank }
   | element JustLastX { s_ST_TrueFalseBlank }
   | element SecretEdit { s_ST_TrueFalseBlank }
   | element Default { s_ST_TrueFalseBlank }
   | element Help { s_ST_TrueFalseBlank }
   | element Cancel { s_ST_TrueFalseBlank }
   | element Dismiss { s_ST_TrueFalseBlank }
   | element Accel { xsd:integer }
   | element Accel2 { xsd:integer }
   | element Row { xsd:integer }
   | element Column { xsd:integer }
   | element Visible { s_ST_TrueFalseBlank }
   | element RowHidden { s_ST_TrueFalseBlank }
   | element ColHidden { s_ST_TrueFalseBlank }
   | element VTEdit { xsd:integer }
   | element MultiLine { s_ST_TrueFalseBlank }
   | element VScroll { s_ST_TrueFalseBlank }
```

```
    | element ValidIds { s_ST_TrueFalseBlank }
    | element FmlaRange { xsd:string }
    | element WidthMin { xsd:integer }
    | element Sel { xsd:integer }
    | element NoThreeD2 { s_ST_TrueFalseBlank }
    | element SelType { xsd:string }
    | element MultiSel { xsd:string }
    | element LCT { xsd:string }
    | element ListItem { xsd:string }
    | element DropStyle { xsd:string }
    | element Colored { s_ST_TrueFalseBlank }
    | element DropLines { xsd:integer }
    | element Checked { xsd:integer }
    | element FmlaLink { xsd:string }
    | element FmlaPict { xsd:string }
    | element NoThreeD { s_ST_TrueFalseBlank }
    | element FirstButton { s_ST_TrueFalseBlank }
    | element FmlaGroup { xsd:string }
    | element Val { xsd:integer }
    | element Min { xsd:integer }
    | element Max { xsd:integer }
    | element Inc { xsd:integer }
    | element Page { xsd:integer }
    | element Horiz { s_ST_TrueFalseBlank }
    | element Dx { xsd:integer }
    | element MapOCX { s_ST_TrueFalseBlank }
    | element CF { x_ST_CF }
    | element Camera { s_ST_TrueFalseBlank }
    | element RecalcAlways { s_ST_TrueFalseBlank }
    | element AutoScale { s_ST_TrueFalseBlank }
    | element DDE { s_ST_TrueFalseBlank }
    | element UIObj { s_ST_TrueFalseBlank }
    | element ScriptText { xsd:string }
    | element ScriptExtended { xsd:string }
    | element ScriptLanguage { xsd:nonNegativeInteger }
    | element ScriptLocation { xsd:nonNegativeInteger }
    | element FmlaTxbx { xsd:string })*
x_ST_CF = xsd:string
x_ST_ObjectType =
  string "Button"
  | string "Checkbox"
  | string "Dialog"
  | string "Drop"
  | string "Edit"
  | string "GBox"
  | string "Label"
  | string "LineA"
  | string "List"
  | string "Movie"
  | string "Note"
  | string "Pict"
  | string "Radio"
  | string "RectA"
```



```
  | string "Scroll"
  | string "Spin"
  | string "Shape"
  | string "Group"
  | string "Rect"
```

B.6.5 VML - Presentation Drawing

This schema is available in the file vml-presentationDrawing.rnc.

```
default namespace = "urn:schemas-microsoft-com:office:powerpoint"
namespace o = "urn:schemas-microsoft-com:office:office"
namespace pvml = "urn:schemas-microsoft-com:office:powerpoint"
namespace v = "urn:schemas-microsoft-com:vml"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"

pvml_iscomment = element iscomment { pvml_CT_Empty }
pvml_textdata = element textdata { pvml_CT_Rel }
pvml_CT_Empty = empty
pvml_CT_Rel = attribute id { xsd:string }?
```

B.6.6 Part Schemas

This schema is available in the file VML_Drawing.rnc.

```
include "wml.rnc"
include "shared-relationshipReference.rnc"
include "dml-wordprocessingDrawing.rnc"
include "dml-main.rnc"
include "dml-diagram.rnc"
include "shared-commonSimpleTypes.rnc"
include "dml-lockedCanvas.rnc"
include "any.rnc"
include "dml-chart.rnc"
include "dml-chartDrawing.rnc"
include "dml-picture.rnc"
include "vml-presentationDrawing.rnc"
include "xml.rnc"
include "shared-customXmlSchemaProperties.rnc"
include "vml-officeDrawing.rnc"
include "vml-main.rnc"
include "vml-spreadsheetDrawing.rnc"
include "vml-wordprocessingDrawing.rnc"
include "shared-math.rnc"
start = element xml {(vml-main | vml-officeDrawing | vml-spreadsheetDrawing |
  vml-presentationDrawing)* }

vml-main =
 v_shape
 | v_shapetype
 | v_group
 | v_background
```

```
  | v_fill
  | v_formulas
  | v_handles
  | v_imagedata
  | v_path
  | v_textbox
  | v_shadow
  | v_stroke
  | v_textpath
  | v_arc
  | v_curve
  | v_image
  | v_line
  | v_oval
  | v_polyline
  | v_rect
  | v_roundrect

vml-officeDrawing =
  o_shapedefaults
   | o_shapelayout
   | o_signatureline
   | o_ink
   | o_diagram
   | o_equationxml
   | o_skew
   | o_extrusion
   | o_callout
   | o_lock
   | o_OLEObject
   | o_complex
   | o_left
   | o_top
   | o_right
   | o_bottom
   | o_column
   | o_clippath
   | o_fill

vml-wordprocessingDrawing =
  w10_bordertop
   | w10_borderleft
   | w10_borderright
   | w10_borderbottom
   | w10_wrap
   | w10_anchorlock

vml-spreadsheetDrawing = x_ClientData
vml-presentationDrawing = pvml_iscomment | pvml_textdata
```


B.7 Shared MLs

B.7.1 Math

This schema is available in the file shared-math.rnc.

```
default namespace m =
  "http://schemas.openxmlformats.org/officeDocument/2006/math"
namespace o = "urn:schemas-microsoft-com:office:office"
namespace s =
  "http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
namespace v = "urn:schemas-microsoft-com:vml"
namespace w =
  "http://schemas.openxmlformats.org/wordprocessingml/2006/main"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"

m_ST_Integer255 =
  xsd:integer { minInclusive = "1" maxInclusive = "255" }
m_CT_Integer255 = attribute m:val { m_ST_Integer255 }
m_ST_Integer2 = xsd:integer { minInclusive = "-2" maxInclusive = "2" }
m_CT_Integer2 = attribute m:val { m_ST_Integer2 }
m_ST_SpacingRule = xsd:integer { minInclusive = "0" maxInclusive = "4" }
m_CT_SpacingRule = attribute m:val { m_ST_SpacingRule }
m_ST_UnSignedInteger = xsd:unsignedInt
m_CT_UnSignedInteger = attribute m:val { m_ST_UnSignedInteger }
m_ST_Char = xsd:string { maxLength = "1" }
m_CT_Char = attribute m:val { m_ST_Char }
m_CT_OnOff = attribute m:val { s_ST_OnOff }?
m_CT_String = attribute m:val { s_ST_String }?
m_CT_XAlign = attribute m:val { s_ST_XAlign }
m_CT_YAlign = attribute m:val { s_ST_YAlign }
m_ST_Shp = string "centered" | string "match"
m_CT_Shp = attribute m:val { m_ST_Shp }
m_ST_FType = string "bar" | string "skw" | string "lin" | string "noBar"
m_CT_FType = attribute m:val { m_ST_FType }
m_ST_LimLoc = string "undOvr" | string "subSup"
m_CT_LimLoc = attribute m:val { m_ST_LimLoc }
m_ST_TopBot = string "top" | string "bot"
m_CT_TopBot = attribute m:val { m_ST_TopBot }
m_ST_Script =
  string "roman"
  | string "script"
  | string "fraktur"
  | string "double-struck"
  | string "sans-serif"
  | string "monospace"
m_CT_Script = attribute m:val { m_ST_Script }?
m_ST_Style = string "p" | string "b" | string "i" | string "bi"
m_CT_Style = attribute m:val { m_ST_Style }?
m_CT_ManualBreak = attribute m:alnAt { m_ST_Integer255 }?
m_EG_ScriptStyle =
```

```
  element scr { m_CT_Script }?,
  element sty { m_CT_Style }?
m_CT_RPR =
  element lit { m_CT_OnOff }?,
  (element nor { m_CT_OnOff }?
   | m_EG_ScriptStyle),
  element brk { m_CT_ManualBreak }?,
  element aln { m_CT_OnOff }?
m_CT_Text = s_ST_String, xml_space?
m_CT_R =
  element rPr { m_CT_RPR }?,
  w_EG_RPr?,
  (w_EG_RunInnerContent
   | element t { m_CT_Text }?)*
m_CT_CtrlPr = w_EG_RPrMath?
m_CT_AccPr =
  element chr { m_CT_Char }?,
  element ctrlPr { m_CT_CtrlPr }?
m_CT_Acc =
  element accPr { m_CT_AccPr }?,
  element e { m_CT_OMathArg }
m_CT_BarPr =
  element pos { m_CT_TopBot }?,
  element ctrlPr { m_CT_CtrlPr }?
m_CT_Bar =
  element barPr { m_CT_BarPr }?,
  element e { m_CT_OMathArg }
m_CT_BoxPr =
  element opEmu { m_CT_OnOff }?,
  element noBreak { m_CT_OnOff }?,
  element diff { m_CT_OnOff }?,
  element brk { m_CT_ManualBreak }?,
  element aln { m_CT_OnOff }?,
  element ctrlPr { m_CT_CtrlPr }?
m_CT_Box =
  element boxPr { m_CT_BoxPr }?,
  element e { m_CT_OMathArg }
m_CT_BorderBoxPr =
  element hideTop { m_CT_OnOff }?,
  element hideBot { m_CT_OnOff }?,
  element hideLeft { m_CT_OnOff }?,
  element hideRight { m_CT_OnOff }?,
  element strikeH { m_CT_OnOff }?,
  element strikeV { m_CT_OnOff }?,
  element strikeBLTR { m_CT_OnOff }?,
  element strikeTLBR { m_CT_OnOff }?,
  element ctrlPr { m_CT_CtrlPr }?
m_CT_BorderBox =
  element borderBoxPr { m_CT_BorderBoxPr }?,
  element e { m_CT_OMathArg }
m_CT_DPr =
  element begChr { m_CT_Char }?,
  element sepChr { m_CT_Char }?,
```



```
  element endChr { m_CT_Char }?,
  element grow { m_CT_OnOff }?,
  element shp { m_CT_Shp }?,
  element ctrlPr { m_CT_CtrlPr }?
m_CT_D =
  element dPr { m_CT_DPr }?,
  element e { m_CT_OMathArg }+
m_CT_EqArrPr =
  element baseJc { m_CT_YAlign }?,
  element maxDist { m_CT_OnOff }?,
  element objDist { m_CT_OnOff }?,
  element rSpRule { m_CT_SpacingRule }?,
  element rSp { m_CT_UnSignedInteger }?,
  element ctrlPr { m_CT_CtrlPr }?
m_CT_EqArr =
  element eqArrPr { m_CT_EqArrPr }?,
  element e { m_CT_OMathArg }+
m_CT_FPr =
  element type { m_CT_FType }?,
  element ctrlPr { m_CT_CtrlPr }?
m_CT_F =
  element fPr { m_CT_FPr }?,
  element num { m_CT_OMathArg },
  element den { m_CT_OMathArg }
m_CT_FuncPr = element ctrlPr { m_CT_CtrlPr }?
m_CT_Func =
  element funcPr { m_CT_FuncPr }?,
  element fName { m_CT_OMathArg },
  element e { m_CT_OMathArg }
m_CT_GroupChrPr =
  element chr { m_CT_Char }?,
  element pos { m_CT_TopBot }?,
  element vertJc { m_CT_TopBot }?,
  element ctrlPr { m_CT_CtrlPr }?
m_CT_GroupChr =
  element groupChrPr { m_CT_GroupChrPr }?,
  element e { m_CT_OMathArg }
m_CT_LimLowPr = element ctrlPr { m_CT_CtrlPr }?
m_CT_LimLow =
  element limLowPr { m_CT_LimLowPr }?,
  element e { m_CT_OMathArg },
  element lim { m_CT_OMathArg }
m_CT_LimUppPr = element ctrlPr { m_CT_CtrlPr }?
m_CT_LimUpp =
  element limUppPr { m_CT_LimUppPr }?,
  element e { m_CT_OMathArg },
  element lim { m_CT_OMathArg }
m_CT_MCPr =
  element count { m_CT_Integer255 }?,
  element mcJc { m_CT_XAlign }?
m_CT_MC = element mcPr { m_CT_MCPr }?
m_CT_MCS = element mc { m_CT_MC }+
m_CT_MPr =
```

```
  element baseJc { m_CT_YAlign }?,
  element plcHide { m_CT_OnOff }?,
  element rSpRule { m_CT_SpacingRule }?,
  element cGpRule { m_CT_SpacingRule }?,
  element rSp { m_CT_UnSignedInteger }?,
  element cSp { m_CT_UnSignedInteger }?,
  element cGp { m_CT_UnSignedInteger }?,
  element mcs { m_CT_MCS }?,
  element ctrlPr { m_CT_CtrlPr }?
m_CT_MR = element e { m_CT_OMathArg }+
m_CT_M =
  element mPr { m_CT_MPr }?,
  element mr { m_CT_MR }+
m_CT_NaryPr =
  element chr { m_CT_Char }?,
  element limLoc { m_CT_LimLoc }?,
  element grow { m_CT_OnOff }?,
  element subHide { m_CT_OnOff }?,
  element supHide { m_CT_OnOff }?,
  element ctrlPr { m_CT_CtrlPr }?
m_CT_Nary =
  element naryPr { m_CT_NaryPr }?,
  element sub { m_CT_OMathArg },
  element sup { m_CT_OMathArg },
  element e { m_CT_OMathArg }
m_CT_PhantPr =
  element show { m_CT_OnOff }?,
  element zeroWid { m_CT_OnOff }?,
  element zeroAsc { m_CT_OnOff }?,
  element zeroDesc { m_CT_OnOff }?,
  element transp { m_CT_OnOff }?,
  element ctrlPr { m_CT_CtrlPr }?
m_CT_Phant =
  element phantPr { m_CT_PhantPr }?,
  element e { m_CT_OMathArg }
m_CT_RadPr =
  element degHide { m_CT_OnOff }?,
  element ctrlPr { m_CT_CtrlPr }?
m_CT_Rad =
  element radPr { m_CT_RadPr }?,
  element deg { m_CT_OMathArg },
  element e { m_CT_OMathArg }
m_CT_SPrePr = element ctrlPr { m_CT_CtrlPr }?
m_CT_SPre =
  element sPrePr { m_CT_SPrePr }?,
  element sub { m_CT_OMathArg },
  element sup { m_CT_OMathArg },
  element e { m_CT_OMathArg }
m_CT_SSubPr = element ctrlPr { m_CT_CtrlPr }?
m_CT_SSub =
  element sSubPr { m_CT_SSubPr }?,
  element e { m_CT_OMathArg },
  element sub { m_CT_OMathArg }
```



```
m_CT_SSubSupPr =
  element alnScr { m_CT_OnOff }?,
  element ctrlPr { m_CT_CtrlPr }?
m_CT_SSubSup =
  element sSubSupPr { m_CT_SSubSupPr }?,
  element e { m_CT_OMathArg },
  element sub { m_CT_OMathArg },
  element sup { m_CT_OMathArg }
m_CT_SSupPr = element ctrlPr { m_CT_CtrlPr }?
m_CT_SSup =
  element sSupPr { m_CT_SSupPr }?,
  element e { m_CT_OMathArg },
  element sup { m_CT_OMathArg }
m_EG_OMathMathElements =
  element acc { m_CT_Acc }
  | element bar { m_CT_Bar }
  | element box { m_CT_Box }
  | element borderBox { m_CT_BorderBox }
  | element d { m_CT_D }
  | element eqArr { m_CT_EqArr }
  | element f { m_CT_F }
  | element func { m_CT_Func }
  | element groupChr { m_CT_GroupChr }
  | element limLow { m_CT_LimLow }
  | element limUpp { m_CT_LimUpp }
  | element m { m_CT_M }
  | element nary { m_CT_Nary }
  | element phant { m_CT_Phant }
  | element rad { m_CT_Rad }
  | element sPre { m_CT_SPre }
  | element sSub { m_CT_SSub }
  | element sSubSup { m_CT_SSubSup }
  | element sSup { m_CT_SSup }
  | element r { m_CT_R }
m_EG_OMathElements = m_EG_OMathMathElements | w_EG_PContentMath
m_CT_OMathArgPr = element argSz { m_CT_Integer2 }?
m_CT_OMathArg =
  element argPr { m_CT_OMathArgPr }?,
  m_EG_OMathElements*,
  element ctrlPr { m_CT_CtrlPr }?
m_ST_Jc =
  string "left"
  | string "right"
  | string "center"
  | string "centerGroup"
m_CT_OMathJc = attribute m:val { m_ST_Jc }?
m_CT_OMathParaPr = element jc { m_CT_OMathJc }?
m_CT_TwipsMeasure = attribute m:val { s_ST_TwipsMeasure }
m_ST_BreakBin = string "before" | string "after" | string "repeat"
m_CT_BreakBin = attribute m:val { m_ST_BreakBin }?
m_ST_BreakBinSub = string "--" | string "-+" | string "+-"
m_CT_BreakBinSub = attribute m:val { m_ST_BreakBinSub }?
m_CT_MathPr =
```

```
  element mathFont { m_CT_String }?,
  element brkBin { m_CT_BreakBin }?,
  element brkBinSub { m_CT_BreakBinSub }?,
  element smallFrac { m_CT_OnOff }?,
  element dispDef { m_CT_OnOff }?,
  element lMargin { m_CT_TwipsMeasure }?,
  element rMargin { m_CT_TwipsMeasure }?,
  element defJc { m_CT_OMathJc }?,
  element preSp { m_CT_TwipsMeasure }?,
  element postSp { m_CT_TwipsMeasure }?,
  element interSp { m_CT_TwipsMeasure }?,
  element intraSp { m_CT_TwipsMeasure }?,
  (element wrapIndent { m_CT_TwipsMeasure }
   | element wrapRight { m_CT_OnOff })?,
  element intLim { m_CT_LimLoc }?,
  element naryLim { m_CT_LimLoc }?
m_mathPr = element mathPr { m_CT_MathPr }
m_CT_OMathPara =
  element oMathParaPr { m_CT_OMathParaPr }?,
  element oMath { m_CT_OMath }+
m_CT_OMath = m_EG_OMathElements*
m_oMathPara = element oMathPara { m_CT_OMathPara }
m_oMath = element oMath { m_CT_OMath }
```

B.7.2 Extended Properties

This schema is available in the file shared-documentPropertiesExtended.rnc.

```
default namespace =
  "http://schemas.openxmlformats.org/officeDocument/2006/extended-properties"
namespace o = "urn:schemas-microsoft-com:office:office"
namespace shdDcEP =
  "http://schemas.openxmlformats.org/officeDocument/2006/extended-properties"
namespace v = "urn:schemas-microsoft-com:vml"
namespace vt =
  "http://schemas.openxmlformats.org/officeDocument/2006/docPropsVTypes"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"

shdDcEP_Properties = element Properties { shdDcEP_CT_Properties }
shdDcEP_CT_Properties =
  element Template { xsd:string }?
  & element Manager { xsd:string }?
  & element Company { xsd:string }?
  & element Pages { xsd:int }?
  & element Words { xsd:int }?
  & element Characters { xsd:int }?
  & element PresentationFormat { xsd:string }?
  & element Lines { xsd:int }?
  & element Paragraphs { xsd:int }?
  & element Slides { xsd:int }?
  & element Notes { xsd:int }?
  & element TotalTime { xsd:int }?
```



```
  & element HiddenSlides { xsd:int }?
  & element MMClips { xsd:int }?
  & element ScaleCrop { xsd:boolean }?
  & element HeadingPairs { shdDcEP_CT_VectorVariant }?
  & element TitlesOfParts { shdDcEP_CT_VectorLpstr }?
  & element LinksUpToDate { xsd:boolean }?
  & element CharactersWithSpaces { xsd:int }?
  & element SharedDoc { xsd:boolean }?
  & element HyperlinkBase { xsd:string }?
  & element HLinks { shdDcEP_CT_VectorVariant }?
  & element HyperlinksChanged { xsd:boolean }?
  & element DigSig { shdDcEP_CT_DigSigBlob }?
  & element Application { xsd:string }?
  & element AppVersion { xsd:string }?
  & element DocSecurity { xsd:int }?
shdDcEP_CT_VectorVariant = vt_vector
shdDcEP_CT_VectorLpstr = vt_vector
shdDcEP_CT_DigSigBlob = vt_blob
```

B.7.2.1 Part Schemas

B.7.2.1.1 Extended File Properties Part

This schema is available in the file Shared_Extended_File_Properties.rnc.

```
include "shared-documentPropertiesExtended.rnc"
include "shared-documentPropertiesVariantTypes.rnc"
include "shared-commonSimpleTypes.rnc"
start = shdDcEP_Properties
```

B.7.3 Custom Properties

This schema is available in the file shared-documentPropertiesCustom.rnc.

```
default namespace =
  "http://schemas.openxmlformats.org/officeDocument/2006/custom-properties"
namespace o = "urn:schemas-microsoft-com:office:office"
namespace s =
  "http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
namespace shdCstm =
  "http://schemas.openxmlformats.org/officeDocument/2006/custom-properties"
namespace v = "urn:schemas-microsoft-com:vml"
namespace vt =
  "http://schemas.openxmlformats.org/officeDocument/2006/docPropsVTypes"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"

shdCstm_Properties = element Properties { shdCstm_CT_Properties }
shdCstm_CT_Properties = element property { shdCstm_CT_Property }*
shdCstm_CT_Property =
  attribute fmtid { s_ST_Guid },
  attribute pid { xsd:int },
  attribute name { xsd:string }?,
  attribute linkTarget { xsd:string }?,
```

```
  (vt_vector
   | vt_array
   | vt_blob
   | vt_oblob
   | vt_empty
   | vt_null
   | vt_i1
   | vt_i2
   | vt_i4
   | vt_i8
   | vt_int
   | vt_ui1
   | vt_ui2
   | vt_ui4
   | vt_ui8
   | vt_uint
   | vt_r4
   | vt_r8
   | vt_decimal
   | vt_lpstr
   | vt_lpwstr
   | vt_bstr
   | vt_date
   | vt_filetime
   | vt_bool
   | vt_cy
   | vt_error
   | vt_stream
   | vt_ostream
   | vt_storage
   | vt_ostorage
   | vt_vstream
   | vt_clsid)
```

B.7.3.1 Part Schemas

B.7.3.1.1 Custom File Properties Part

This schema is available in the file Shared_Custom_File_Properties.rnc.

```
include "shared-documentPropertiesCustom.rnc"
include "shared-documentPropertiesVariantTypes.rnc"
include "shared-commonSimpleTypes.rnc"
start = shdCstm_Properties
```

B.7.4 Variant Types

This schema is available in the file shared-documentPropertiesVariantTypes.rnc.

```
default namespace =
  "http://schemas.openxmlformats.org/officeDocument/2006/docPropsVTypes"
namespace o = "urn:schemas-microsoft-com:office:office"
namespace s =
  "http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
```



```
namespace v = "urn:schemas-microsoft-com:vml"
namespace vt =
  "http://schemas.openxmlformats.org/officeDocument/2006/docPropsVTypes"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"

vt_ST_VectorBaseType =
  string "variant"
  | string "i1"
  | string "i2"
  | string "i4"
  | string "i8"
  | string "ui1"
  | string "ui2"
  | string "ui4"
  | string "ui8"
  | string "r4"
  | string "r8"
  | string "lpstr"
  | string "lpwstr"
  | string "bstr"
  | string "date"
  | string "filetime"
  | string "bool"
  | string "cy"
  | string "error"
  | string "clsid"
vt_ST_ArrayBaseType =
  string "variant"
  | string "i1"
  | string "i2"
  | string "i4"
  | string "int"
  | string "ui1"
  | string "ui2"
  | string "ui4"
  | string "uint"
  | string "r4"
  | string "r8"
  | string "decimal"
  | string "bstr"
  | string "date"
  | string "bool"
  | string "cy"
  | string "error"
vt_ST_Cy = xsd:string { pattern = "\s*[0-9]*\.[0-9]{4}\s*" }
vt_ST_Error = xsd:string { pattern = "\s*0x[0-9A-Za-z]{8}\s*" }
vt_CT_Empty = empty
vt_CT_Null = empty
vt_CT_Vector =
  attribute baseType { vt_ST_VectorBaseType },
  attribute size { xsd:unsignedInt },
  (vt_variant
```

```
    | vt_i1
    | vt_i2
    | vt_i4
    | vt_i8
    | vt_ui1
    | vt_ui2
    | vt_ui4
    | vt_ui8
    | vt_r4
    | vt_r8
    | vt_lpstr
    | vt_lpwstr
    | vt_bstr
    | vt_date
    | vt_filetime
    | vt_bool
    | vt_cy
    | vt_error
    | vt_clsid)+
vt_CT_Array =
  attribute lBounds { xsd:int },
  attribute uBounds { xsd:int },
  attribute baseType { vt_ST_ArrayBaseType },
  (vt_variant
    | vt_i1
    | vt_i2
    | vt_i4
    | vt_int
    | vt_ui1
    | vt_ui2
    | vt_ui4
    | vt_uint
    | vt_r4
    | vt_r8
    | vt_decimal
    | vt_bstr
    | vt_date
    | vt_bool
    | vt_error
    | vt_cy)+
vt_CT_Variant =
  vt_variant
  | vt_vector
  | vt_array
  | vt_blob
  | vt_oblob
  | vt_empty
  | vt_null
  | vt_i1
  | vt_i2
  | vt_i4
  | vt_i8
  | vt_int
```



```
  | vt_ui1
  | vt_ui2
  | vt_ui4
  | vt_ui8
  | vt_uint
  | vt_r4
  | vt_r8
  | vt_decimal
  | vt_lpstr
  | vt_lpwstr
  | vt_bstr
  | vt_date
  | vt_filetime
  | vt_bool
  | vt_cy
  | vt_error
  | vt_stream
  | vt_ostream
  | vt_storage
  | vt_ostorage
  | vt_vstream
  | vt_clsid
vt_CT_Vstream =
  xsd:base64Binary,
  attribute version { s_ST_Guid }?
vt_variant = element variant { vt_CT_Variant }
vt_vector = element vector { vt_CT_Vector }
vt_array = element array { vt_CT_Array }
vt_blob = element blob { xsd:base64Binary }
vt_oblob = element oblob { xsd:base64Binary }
vt_empty = element empty { vt_CT_Empty }
vt_null = element null { vt_CT_Null }
vt_i1 = element i1 { xsd:byte }
vt_i2 = element i2 { xsd:short }
vt_i4 = element i4 { xsd:int }
vt_i8 = element i8 { xsd:long }
vt_int = element int { xsd:int }
vt_ui1 = element ui1 { xsd:unsignedByte }
vt_ui2 = element ui2 { xsd:unsignedShort }
vt_ui4 = element ui4 { xsd:unsignedInt }
vt_ui8 = element ui8 { xsd:unsignedLong }
vt_uint = element uint { xsd:unsignedInt }
vt_r4 = element r4 { xsd:float }
vt_r8 = element r8 { xsd:double }
vt_decimal = element decimal { xsd:decimal }
vt_lpstr = element lpstr { xsd:string }
vt_lpwstr = element lpwstr { xsd:string }
vt_bstr = element bstr { xsd:string }
vt_date = element date { xsd:dateTime }
vt_filetime = element filetime { xsd:dateTime }
vt_bool = element bool { xsd:boolean }
vt_cy = element cy { vt_ST_Cy }
vt_error = element error { vt_ST_Error }
```

```
vt_stream = element stream { xsd:base64Binary }
vt_ostream = element ostream { xsd:base64Binary }
vt_storage = element storage { xsd:base64Binary }
vt_ostorage = element ostorage { xsd:base64Binary }
vt_vstream = element vstream { vt_CT_Vstream }
vt_clsid = element clsid { s_ST_Guid }
```

B.7.5 Custom XML Data Properties

This schema is available in the file shared-customXmlDataProperties.rnc.

```
default namespace ds =
  "http://schemas.openxmlformats.org/officeDocument/2006/customXml"
namespace o = "urn:schemas-microsoft-com:office:office"
namespace s =
  "http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
namespace v = "urn:schemas-microsoft-com:vml"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"

ds_CT_DatastoreSchemaRef = attribute ds:uri { xsd:string }
ds_CT_DatastoreSchemaRefs =
  element schemaRef { ds_CT_DatastoreSchemaRef }*
ds_CT_DatastoreItem =
  attribute ds:itemID { s_ST_Guid },
  element schemaRefs { ds_CT_DatastoreSchemaRefs }?
ds_datastoreItem = element datastoreItem { ds_CT_DatastoreItem }
```

B.7.5.1 Part Schemas

B.7.5.1.1 Custom XML Data Properties Part

This schema is available in the file Shared_Custom_XML_Data_Storage_Properties.rnc.

```
include "shared-customXmlDataProperties.rnc"
include "shared-commonSimpleTypes.rnc"
start = ds_datastoreItem
```

B.7.6 Bibliography

This schema is available in the file shared-bibliography.rnc.

```
default namespace =
  "http://schemas.openxmlformats.org/officeDocument/2006/bibliography"
namespace o = "urn:schemas-microsoft-com:office:office"
namespace s =
  "http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
namespace shrdBib =
  "http://schemas.openxmlformats.org/officeDocument/2006/bibliography"
namespace v = "urn:schemas-microsoft-com:vml"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"

shrdBib_ST_SourceType =
```



```
  "ArticleInAPeriodical"
  | "Book"
  | "BookSection"
  | "JournalArticle"
  | "ConferenceProceedings"
  | "Report"
  | "SoundRecording"
  | "Performance"
  | "Art"
  | "DocumentFromInternetSite"
  | "InternetSite"
  | "Film"
  | "Interview"
  | "Patent"
  | "ElectronicSource"
  | "Case"
  | "Misc"
shrdBib_CT_NameListType = element Person { shrdBib_CT_PersonType }+
shrdBib_CT_PersonType =
  element Last { s_ST_String }*,
  element First { s_ST_String }*,
  element Middle { s_ST_String }*
shrdBib_CT_NameType = element NameList { shrdBib_CT_NameListType }
shrdBib_CT_NameOrCorporateType =
  (element NameList { shrdBib_CT_NameListType }
   | element Corporate { s_ST_String })?
shrdBib_CT_AuthorType =
  (element Artist { shrdBib_CT_NameType }
   | element Author { shrdBib_CT_NameOrCorporateType }
   | element BookAuthor { shrdBib_CT_NameType }
   | element Compiler { shrdBib_CT_NameType }
   | element Composer { shrdBib_CT_NameType }
   | element Conductor { shrdBib_CT_NameType }
   | element Counsel { shrdBib_CT_NameType }
   | element Director { shrdBib_CT_NameType }
   | element Editor { shrdBib_CT_NameType }
   | element Interviewee { shrdBib_CT_NameType }
   | element Interviewer { shrdBib_CT_NameType }
   | element Inventor { shrdBib_CT_NameType }
   | element Performer { shrdBib_CT_NameOrCorporateType }
   | element ProducerName { shrdBib_CT_NameType }
   | element Translator { shrdBib_CT_NameType }
   | element Writer { shrdBib_CT_NameType })*
shrdBib_CT_SourceType =
  (element AbbreviatedCaseNumber { s_ST_String }
   | element AlbumTitle { s_ST_String }
   | element Author { shrdBib_CT_AuthorType }
   | element BookTitle { s_ST_String }
   | element Broadcaster { s_ST_String }
   | element BroadcastTitle { s_ST_String }
   | element CaseNumber { s_ST_String }
   | element ChapterNumber { s_ST_String }
   | element City { s_ST_String }
```

```
    | element Comments { s_ST_String }
    | element ConferenceName { s_ST_String }
    | element CountryRegion { s_ST_String }
    | element Court { s_ST_String }
    | element Day { s_ST_String }
    | element DayAccessed { s_ST_String }
    | element Department { s_ST_String }
    | element Distributor { s_ST_String }
    | element Edition { s_ST_String }
    | element Guid { s_ST_String }
    | element Institution { s_ST_String }
    | element InternetSiteTitle { s_ST_String }
    | element Issue { s_ST_String }
    | element JournalName { s_ST_String }
    | element LCID { s_ST_Lang }
    | element Medium { s_ST_String }
    | element Month { s_ST_String }
    | element MonthAccessed { s_ST_String }
    | element NumberVolumes { s_ST_String }
    | element Pages { s_ST_String }
    | element PatentNumber { s_ST_String }
    | element PeriodicalTitle { s_ST_String }
    | element ProductionCompany { s_ST_String }
    | element PublicationTitle { s_ST_String }
    | element Publisher { s_ST_String }
    | element RecordingNumber { s_ST_String }
    | element RefOrder { s_ST_String }
    | element Reporter { s_ST_String }
    | element SourceType { shrdBib_ST_SourceType }
    | element ShortTitle { s_ST_String }
    | element StandardNumber { s_ST_String }
    | element StateProvince { s_ST_String }
    | element Station { s_ST_String }
    | element Tag { s_ST_String }
    | element Theater { s_ST_String }
    | element ThesisType { s_ST_String }
    | element Title { s_ST_String }
    | element Type { s_ST_String }
    | element URL { s_ST_String }
    | element Version { s_ST_String }
    | element Volume { s_ST_String }
    | element Year { s_ST_String }
    | element YearAccessed { s_ST_String })*
shrdBib_Sources = element Sources { shrdBib_CT_Sources }
shrdBib_CT_Sources =
  attribute SelectedStyle { s_ST_String }?,
  attribute StyleName { s_ST_String }?,
  attribute URI { s_ST_String }?,
  element Source { shrdBib_CT_SourceType }*
```


B.7.6.1 Part Schemas

B.7.6.1.1 Bibliography Part

This schema is available in the file Shared_Bibliography.rnc.

```
include "shared-bibliography.rnc"
include "shared-commonSimpleTypes.rnc"
start = shrdBib_Sources
```

B.7.7 Additional Characteristics

This schema is available in the file shared-additionalCharacteristics.rnc.

```
default namespace =
  "http://schemas.openxmlformats.org/officeDocument/2006/characteristics"
namespace o = "urn:schemas-microsoft-com:office:office"
namespace shrdChr =
  "http://schemas.openxmlformats.org/officeDocument/2006/characteristics"
namespace v = "urn:schemas-microsoft-com:vml"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"

shrdChr_CT_AdditionalCharacteristics =
  element characteristic { shrdChr_CT_Characteristic }*
shrdChr_CT_Characteristic =
  attribute name { xsd:string },
  attribute relation { shrdChr_ST_Relation },
  attribute val { xsd:string },
  attribute vocabulary { xsd:anyURI }?
shrdChr_ST_Relation =
  string "ge" | string "le" | string "gt" | string "lt" | string "eq"
shrdChr_additionalCharacteristics =
  element additionalCharacteristics {
    shrdChr_CT_AdditionalCharacteristics
  }
```

B.7.7.1 Part Schemas

B.7.7.1.1 Additional Characteristics Part

This schema is available in the file Shared_Additional_Characteristics.rnc.

```
include "shared-additionalCharacteristics.rnc"
start = shrdChr_additionalCharacteristics
```

B.7.8 Office Document Relationships

This schema is available in the file shared-relationshipReference.rnc.

```
namespace o = "urn:schemas-microsoft-com:office:office"
namespace r =
  "http://schemas.openxmlformats.org/officeDocument/2006/relationships"
```

```
namespace v = "urn:schemas-microsoft-com:vml"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"

r_ST_RelationshipId = xsd:string
r_id = attribute r:id { r_ST_RelationshipId }
r_embed = attribute r:embed { r_ST_RelationshipId }
r_link = attribute r:link { r_ST_RelationshipId }
r_dm = attribute r:dm { r_ST_RelationshipId }
r_lo = attribute r:lo { r_ST_RelationshipId }
r_qs = attribute r:qs { r_ST_RelationshipId }
r_cs = attribute r:cs { r_ST_RelationshipId }
r_blip = attribute r:blip { r_ST_RelationshipId }
r_pict = attribute r:pict { r_ST_RelationshipId }
r_href = attribute r:href { r_ST_RelationshipId }
r_topLeft = attribute r:topLeft { r_ST_RelationshipId }
r_topRight = attribute r:topRight { r_ST_RelationshipId }
r_bottomLeft = attribute r:bottomLeft { r_ST_RelationshipId }
r_bottomRight = attribute r:bottomRight { r_ST_RelationshipId }
```

B.7.9 Shared Simple Types

This schema is available in the file shared-commonSimpleTypes.rnc.

```
namespace o = "urn:schemas-microsoft-com:office:office"
namespace s =
  "http://schemas.openxmlformats.org/officeDocument/2006/sharedTypes"
namespace v = "urn:schemas-microsoft-com:vml"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"

s_ST_Lang = xsd:string
s_ST_HexColorRGB = xsd:hexBinary { length = "3" }
s_ST_Panose = xsd:hexBinary { length = "10" }
s_ST_CalendarType =
  string "gregorian"
  | string "gregorianUs"
  | string "gregorianMeFrench"
  | string "gregorianArabic"
  | string "hijri"
  | string "hebrew"
  | string "taiwan"
  | string "japan"
  | string "thai"
  | string "korea"
  | string "saka"
  | string "gregorianXlitEnglish"
  | string "gregorianXlitFrench"
  | string "none"
s_ST_AlgClass = string "hash" | string "custom"
s_ST_CryptProv = string "rsaAES" | string "rsaFull" | string "custom"
s_ST_AlgType = string "typeAny" | string "custom"
s_ST_ColorType = xsd:string
```



```
s_ST_Guid =
  xsd:token {
    pattern =
      "\{[0-9A-F]{8}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{12}\}"
  }
s_ST_OnOff = xsd:boolean | s_ST_OnOff1
s_ST_OnOff1 = string "on" | string "off"
s_ST_String = xsd:string
s_ST_XmlName = xsd:NCName { minLength = "1" maxLength = "255" }
s_ST_TrueFalse =
  string "t" | string "f" | string "true" | string "false"
s_ST_TrueFalseBlank =
  string "t"
  | string "f"
  | string "true"
  | string "false"
  | string ""
  | string "True"
  | string "False"
s_ST_UnsignedDecimalNumber = xsd:unsignedLong
s_ST_TwipsMeasure =
  s_ST_UnsignedDecimalNumber | s_ST_PositiveUniversalMeasure
s_ST_VerticalAlignRun =
  string "baseline" | string "superscript" | string "subscript"
s_ST_Xstring = xsd:string
s_ST_XAlign =
  string "left"
  | string "center"
  | string "right"
  | string "inside"
  | string "outside"
s_ST_YAlign =
  string "inline"
  | string "top"
  | string "center"
  | string "bottom"
  | string "inside"
  | string "outside"
s_ST_ConformanceClass = string "strict" | string "transitional"
s_ST_UniversalMeasure =
  xsd:string { pattern = "-?[0-9]+(\.[0-9]+)?(mm|cm|in|pt|pc|pi)" }
s_ST_PositiveUniversalMeasure =
  xsd:string {
    pattern = "-?[0-9]+(\.[0-9]+)?(mm|cm|in|pt|pc|pi)"
    pattern = "[0-9]+(\.[0-9]+)?(mm|cm|in|pt|pc|pi)"
  }
s_ST_Percentage = xsd:string { pattern = "-?[0-9]+(\.[0-9]+)?%" }
s_ST_FixedPercentage =
  xsd:string {
    pattern = "-?[0-9]+(\.[0-9]+)?%"
    pattern = "-?((100)|([0-9][0-9]?))(\.[0-9][0-9]?)?%"
  }
s_ST_PositivePercentage =
```

```
  xsd:string {
    pattern = "-?[0-9]+(\.[0-9]+)?%"
    pattern = "[0-9]+(\.[0-9]+)?%"
  }
s_ST_PositiveFixedPercentage =
  xsd:string {
    pattern = "-?[0-9]+(\.[0-9]+)?%"
    pattern = "((100)|([0-9][0-9]?))(\.[0-9][0-9]?)?%"
  }
```

B.8 Custom XML Schema References

This schema is available in the file shared-customXmlSchemaProperties.rnc.

```
namespace o = "urn:schemas-microsoft-com:office:office"
default namespace sl =
  "http://schemas.openxmlformats.org/schemaLibrary/2006/main"
namespace v = "urn:schemas-microsoft-com:vml"
namespace w10 = "urn:schemas-microsoft-com:office:word"
namespace x = "urn:schemas-microsoft-com:office:excel"

sl_CT_Schema =
  attribute sl:uri { xsd:string }?,
  attribute sl:manifestLocation { xsd:string }?,
  attribute sl:schemaLocation { xsd:string }?,
  attribute sl:schemaLanguage { xsd:token }?
sl_CT_SchemaLibrary = element schema { sl_CT_Schema }*
sl_schemaLibrary = element schemaLibrary { sl_CT_SchemaLibrary }
```

B.9 Additional Resources

B.9.1 Any

This schema is available in the file any.rnc.

```
anyElement = element * { anyAttribute*, text?, anyElement* }
anyAttribute = attribute * { text }
```

B.9.2 XML

This schema is available in the file xml.rnc.

```
xml_lang = attribute xml:lang { xsd:language | xsd:string "" }
xml_space = attribute xml:space { "default" | "preserve" }
xml_base = attribute xml:base { xsd:anyURI }
xml_id = attribute xml:id { xsd:ID }
xml_specialAttrs = xml_base?, xml_lang?, xml_space?, xml_id?
```

End of informative text.

Annex C.
(informative)
Namespace Prefix Mapping in Examples

This Annex is informative.

Throughout ISO/IEC 29500, XML syntax is provided to illustrate the concepts being documented. These examples leverage XML namespace prefixes, and, typically, for brevity, do not show the actual namespace mappings. This Annex lists the namespace prefix mappings that are used within these examples.

Prefix	Namespace
a	http://schemas.openxmlformats.org/drawingml/2006/main
b	http://schemas.openxmlformats.org/officeDocument/2006/bibliography
cp	http://schemas.openxmlformats.org/package/2006/metadata/core-properties
cdr	http://schemas.openxmlformats.org/drawingml/2006/chartDrawing
dc	http://purl.org/dc/elements/1.1/
dcmitype	http://purl.org/dc/dcmitype/
dcterms	http://purl.org/dc/terms/
ds	http://schemas.openxmlformats.org/officeDocument/2006/customXml
m	http://schemas.openxmlformats.org/officeDocument/2006/math
o	urn:schemas-microsoft-com:office:office
p	http://schemas.openxmlformats.org/presentationml/2006/main
pic	http://schemas.openxmlformats.org/drawingml/2006/picture
pvml	urn:schemas-microsoft-com:office:powerpoint
r	http://schemas.openxmlformats.org/officeDocument/2006/relationships
sl	http://schemas.openxmlformats.org/schemaLibrary/2006/main
v	urn:schemas-microsoft-com:vml
ve	http://schemas.openxmlformats.org/markup-compatibility/2006
vt	http://schemas.openxmlformats.org/officeDocument/2006/docPropsVTypes
w	http://schemas.openxmlformats.org/wordprocessingml/2006/main
w10	urn:schemas-microsoft-com:office:word

Prefix	Namespace
wp	http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing
x	urn:schemas-microsoft-com:office:excel
xdr	http://schemas.openxmlformats.org/drawingml/2006/spreadsheetDrawing
xsd	http://www.w3.org/2001/XMLSchema
xsi	http://www.w3.org/2001/XMLSchema-instance

If no namespace prefix is specified, it should be assumed that that element or attribute is contained within the namespace defined by the parent subclause. For example, unprefixed elements in Part 1, §18 are contained in the http://schemas.openxmlformats.org/spreadsheetml/2006/main namespace.

End informative Annex.

Annex D.
(informative)
Differences Between ISO/IEC 29500 and ECMA-376:2006

This annex is informative.

This annex highlights the differences between the versions of the Transitional form of the Office Open XML schemas, as defined in ISO/IEC 29500 and the schemas as defined by ECMA-376:2006.

D.1 WordprocessingML

The following changes occurred to the WordprocessingML schema:

- The algorithmName, hashValue, saltValue, and spinCount attributes were added to documentProtection (Part 1, §17.15.1.29) and writeProtection (Part 1, §17.15.1.93)
- The allStyles, customStyles, latentStyles, stylesInUse, headingStyles, numberingStyles, tableStyles, directFormattingOnRuns, directFormattingOnParagraphs, directFormattingOnNumbering, directFormattingOnTables, clearFormatting, top3HeadingStyles, and visibleStyles attributes were added to the stylePaneFormatFilter element (Part 1, §17.15.1.85)
- The bdo element (Part 1, §17.3.2.3) was added
- The characterSet attribute was added to the charset element (Part 1, §17.8.3.2)
- The compatSetting element (Part 1, §17.15.3.4) was added
- The conformance attribute was added to document (Part 1, §17.2.3)
- The content model of ST_HpsMeasure (Part 1, §17.18.42) was modified to allow ST_PositiveUniversalMeasure (Part 1, §22.9.2.12)
- The content model of ST_OnOff (Part 1, §22.9.2.7) was changed to an xsd:boolean, removing the values on and off
- The content model of ST_SignedHpsMeasure (Part 1, §17.18.80) was modified to allow ST_UniversalMeasure (Part 1, §22.9.2.15)
- The content model of ST_SignedTwipsMeasure (Part 1, §17.18.81) was modified to allow ST_UniversalMeasure (Part 1, §22.9.2.15)
- The contentPart element (Part 1, §17.3.3.2) was added
- The dir element (Part 1, §17.3.2.8) was added
- The end element (Part 1, §17.4.10) was added
- The end element (Part 1, §17.4.11) was added
- The end element (Part 1, §17.4.12) was added

- The end element (Part 1, §17.4.13) was added
- The firstRow, lastRow, firstColumn, lastColumn, noHBand, and noVBand attributes were added to the tblLook element (Part 1, §17.4.55; Part 1, §17.4.56)
- The firstRow, lastRow, firstColumn, lastColumn, oddVBand, evenVBand, oddHBand, evenHBand, firstRowFirstColumn, firstRowLastColumn, lastRowFirstColumn, and lastRowLastColumn attributes were added to the cnfStyle element (Part 1, §17.3.1.8; Part 1, §17.4.8; Part 1, §17.4.7)
- The following enumeration values were added to the ST_Border simple type (Part 1, §17.18.2): earth3, triangle1, triangle2, triangleCircle1, triangleCircle2, shapes1, shapes2, custom
- The following enumeration values were added to the ST_CalendarType simple type (Part 1, §22.9.2.1): `gregorianArabic`, `gregorianMeFrench`, and `gregorianUs`.
- The following enumeration values were added to the ST_Jc simple type (Part 1, §17.18.44): `start`, `end`
- The following enumeration values were added to the ST_NumberFormat simple type (Part 1, §17.18.59): `bahtText`, `dollarText`, `custom`
- The following enumeration values were added to the ST_TabJc simple type (Part 1, §17.18.84): `start`, `end`
- The following enumeration values were added to the ST_TextDirection simple type (Part 1, §17.18.93): `tb`, `rl`, `lr`, `tbV`, `rlV`, and `lrV`.
- The following enumeration values were removed from the ST_Border simple type (Part 1, §17.18.2): `tribal1`, `tribal2`, `tribal3`, `tribal4`, `tribal5`, `tribal6`
- The fontSz attribute on the readModeInkLockDown element (Part 1, §17.15.1.66) was modified to use ST_DecimalNumberOrPercent (Part 1, §17.18.11)
- The format attribute was added to the numFmt element (Part 1, §17.9.18)
- The header element (Part 1, §17.4.18) was added
- The headers element (Part 1, §17.4.19) was added
- The id attribute was added to the left element (Part 1, §17.6.7) and right element (Part 1, §17.6.15)
- The id attribute was added to the tc element (Part 1, §17.4.66)
- The id, bottomLeft, and bottomRight attributes were added to the bottom element (Part 1, §17.6.2)
- The id, topLeft, and topRight attributes were added to the top element (Part 1, §17.6.21)
- The jc element (Part 1, §17.4.29) was modified to use the ST_JcTable simple type (Part 1, §17.18.45)
- The label element (Part 1, §17.5.2.19) was added
- The longDesc element (Part 1, §17.15.2.23) was added
- The objectEmbed element (Part1, Part 1, §17.3.3.20) was added
- The objectLink element (Part 1, §17.3.3.21) was added
- The percent attribute on the zoom element (Part 1, §17.15.1.94) was modified to use ST_DecimalNumberOrPercent (Part 1, §17.18.11)
- The ST_ColorSchemeIndex simple type was renamed to ST_WmlColorSchemeIndex (Part 1, §17.18.103)
- The ST_DecimalNumberOrPercent (Part 1, §17.18.11) simple type was added
- The ST_Direction simple type (Part 1, §17.18.12) was added
- The ST_DocType simple type (Part 1, §17.18.19) was modified to allow any xsd:string
- The ST_JcTable simple type (Part 1, §17.18.45) was added

- The ST_LangCode simple type was removed
- The ST_MailMergeDataType simple type (Part 1, §17.18.54) was modified to allow any xsd:string
- The ST_ObjectDrawAspect simple type (Part 1, §17.18.60) was added
- The ST_ObjectUpdateMode simple type (Part 1, §17.18.61) was added
- The ST_StyleSort simple type (Part 1, §17.18.82) was added
- The ST_UnqualifiedPercentage simple type (§14.10.10) was added
- The start element (Part 1, §17.4.34) was added
- The start element (Part 1, §17.4.35) was added
- The start element (Part 1, §17.4.36) was added
- The start element (Part 1, §17.4.37) was added
- The start, startChars, end, endChars attributes were added to the ind element (Part 1, §17.3.1.12)
- The tabIndex element (Part 1, §17.5.2.41) was added
- The target attribute was added to the optimizeForBrowser element (Part 1, §17.15.2.33)
- The tblCaption element (Part 1, §17.4.41) was added
- The tblDescription element (Part 1, §17.4.47) was added
- The title element (Part 1, §17.15.2.43) was added
- The uiCompat97To2003 element was removed
- The vendorID and dllVersions attributes on the activeWritingStyle element (Part 1, §17.15.1.1) was modified to use ST_String (Part 1, §22.9.2.13)

D.2 SpreadsheetML

The following changes occurred to the SpreadsheetML schema:

- The algorithmName, hashValue, saltValue, and spinCount attributes were added to sheetProtection (Part 1, §18.3.1.85; Part 1, §18.3.1.84), protectedRange (Part 1, §18.3.1.71), sheetProtection (Part 1, §18.3.1.85), and fileSharing (Part 1, §18.2.12)
- The anchor element (Part 1, §18.3.1.1) was added
- The characterSet attribute was added to the textPr element (Part 1, §18.13.12) and the webPublishing element (Part 1, §18.2.24)
- The commentPr element (Part 1, §18.7.5) was added
- The conformance attribute was added to the workbook element (Part 1, §18.2.27)
- The controlPr element (Part 1, §18.3.1.20) was added
- The drawingHF element (Part 1, §18.3.1.37) was added
- The end element (Part 1, §18.8.16) was added
- The objectPr element (Part 1, §18.3.1.56) was added
- The paperHeight and paperWidth attributes were added to the pageSetup element (Part 1, §18.3.1.63)
- The paperHeight and paperWidth attributes were added to the pageSetup element (Part 1, §18.3.1.64)
- The refreshedDateIso attribute was added to the pivotCacheDefinition element (Part 1, §18.10.1.67)
- The Schema element (Part 1, §18.16.4) now allows mixed content
- The SchemaLanguage attribute was added to the schema element (Part 1, §18.16.4)
- The securityDescriptor element (Part 1, §18.3.1.77) was added

- The shapeId attribute was added to the comment element (Part 1, §18.7.3)
- The ST_CalendarType simple type (Part 1, §22.9.2.1) now allows an enumeration value of saka
- The ST_CellType simple type (Part 1, §18.18.11) now allows an enumeration value of d
- The ST_FileType simple type (Part 1, §18.18.29) now allows enumeration values of lin and other
- The ST_PivotAreaType simple type (Part 1, §18.18.58) now allows an enumeration value of topEnd
- The ST_TextHAlign simple type (Part 1, §18.18.80) was added
- The ST_TextVAlign simple type (Part 1, §18.18.81) was added
- The ST_XmlDataType simple type (Part 1, §18.18.93) was modified to allow any xsd:string
- The start element (Part 1, §18.8.37) was added
- The startLabels attribute was added to the dataConsolidate element (Part 1, §18.3.1.29)
- The valIso and maxValIso attributes were added to the dynamicFilter element (Part 1, §18.3.2.5)
- The workbookPasswordCharacterSet, revisionsPasswordCharacterSet, revisionsAlgorithmName, revisionsHashValue, revisionsSaltValue, revisionsSpinCount, workbookAlgorithmName, workbookHashValue, workbookSaltValue, and workbookSpinCount attributes were added to the workbookProtection element (Part 1, §18.2.29)

D.3 PresentationML

The following changes occurred to the PresentationML schema:

- The algorithmName, hashValue, saltValue, and spinCount attributes were added to the modifyVerifier element (Part 1, §19.2.1.19)
- The conformance attribute was added to the presentation element (Part 1, §19.2.1.26)
- The contentPart element was added (Part 1, §19.3.1.14)
- The pubBrowser attribute on the htmlPubPr element (§16.2.1.1) was renamed target
- The ST_HtmlPublishWebBrowserSupport simple type was removed and replaced by xsd:string

D.4 DrawingML

D.4.1 DrawingML – Main

The following changes occurred to the DrawingML Main schema:

- The builtIn attribute was removed from the snd element (Part 1, §19.5.68)
- The content model of ST_Coordinate (Part 1, §20.1.10.16) was modified to allow ST_UniversalMeasure (Part 1, §22.9.2.15)
- The content model of ST_Coordinate32 (Part 1, §20.1.10.17) was modified to allow ST_UniversalMeasure (Part 1, §22.9.2.15)
- The content model of ST_FixedPercentage (Part 1, §20.1.10.24) was modified to allow ST_FixedPercentage (Part 1, §22.9.2.3)
- The content model of ST_Percentage (Part 1, §20.1.10.40) was modified to allow ST_Percentage (Part 1, §22.9.2.9)
- The content model of ST_PositiveFixedPercentage (Part 1, §20.1.10.45) was modified to allow ST_PositiveFixedPercentage (Part 1, §22.9.2.10)

- The content model of ST_PositivePercentage (Part 1, §20.1.10.46) was modified to allow ST_PositivePercentage (Part 1, §22.9.2.11)
- The contentType attribute was added to the videoFile (Part 1, §20.1.3.6) and audioFile elements (Part 1, §20.1.3.2)
- The header element (Part 1, §21.1.3.3) was added
- The headers element (Part 1, §21.1.3.4) was added
- The id attribute was added to the tc element (Part 1, §21.1.3.16)
- The rtl element (Part 1, §21.1.2.2.8) was added
- The ST_PresetColorVal simple type (Part 1, §20.1.10.48) now allows enumeration values of: `darkBlue`, `darkCyan`, `darkGoldenrod`, `darkGray`, `darkGrey`, `darkGreen`, `darkKhaki`, `darkMagenta`, `darkOliveGreen`, `darkOrange`, `darkOrchid`, `darkRed`, `darkSalmon`, `darkSeaGreen`, `darkSlateBlue`, `darkSlateGray`, `darkSlateGrey`, `darkTurquoise`, `darkViolet`, `dkGrey`, `dkSlateGrey`, `dimGrey`, `grey`, `lightBlue`, `lightCoral`, `lightCyan`, `lightGoldenrodYellow`, `lightGray`, `lightGrey`, `lightGreen`, `lightPink`, `lightSalmon`, `lightSeaGreen`, `lightSkyBlue`, `lightSlateGray`, `lightSlateGrey`, `lightSteelBlue`, `lightYellow`, `ltGrey`, `ltSlateGrey`, `mediumAquamarine`, `mediumBlue`, `mediumOrchid`, `mediumPurple`, `mediumSeaGreen`, `mediumSlateBlue`, `mediumSpringGreen`, `mediumTurquoise`, `mediumVioletRed`, `slateGrey`
- The ST_TextFontScalePercent simple type was renamed to ST_TextFontScalePercentOrPercentString (Part 1, §20.1.10.67) and modified to allow ST_Percentage (Part 1, §22.9.2.9)
- The ST_TextPoint simple type (Part 1, §20.1.10.74) was modified to allow ST_UniversalMeasure (Part 1, §22.9.2.15)
- The ST_TextSpacingPercent simple type was renamed to ST_TextSpacingPercentOrPercentString (Part 1, §20.1.10.77) and modified to allow ST_Percentage (Part 1, §22.9.2.9)
- The title attribute was added to the cNvPr element (Part 1, §20.1.2.2.8)

D.4.2 DrawingML – Chart

The following changes occurred to the Chart schema:

- The paperHeight and paperWidth attributes were added to the pageSetup element (Part 1, §21.2.2.134)

D.4.3 DrawingML – Diagrams

The following changes occurred to the Diagram schema:

- The ST_HorizontalAlignment simple type was renamed to ST_DiagramHorizontalAlignment (Part 1, §21.4.7.24)
- The ST_TextAlignment simple type was renamed to ST_DiagramTextAlignment (Part 1, §21.4.7.25)

D.4.4 DrawingML – Spreadsheet Drawing

The following changes occurred to the Spreadsheet Drawing schema:

- The contentPart element (Part 1, §20.5.2.12) was added

D.5 VML

D.5.1 VML

The following changes occurred to the VML schema:

- The ST_TrueFalseBlank simple type (§20.1.2.6) now allows enumeration values of `True` and `False`

D.5.2 VML – Office Drawing

The following changes occurred to the Office Drawing schema:

- The ST_DiagramLayout simple type (§19.2.3.10) was added
- The equationxml element (§19.2.2.10) was added
- The contentType attribute was added to the ink element (§19.2.2.15)
- The ST_AlternateMathContentType simple type (§19.2.3.1) was added
- The ST_OLELinkType simple type (§19.2.3.19) was modified to allow any xsd:string
- The ST_TrueFalseBlank simple type (§20.1.2.6) now allows enumeration values of `True` and `False`

D.5.3 VML – Spreadsheet Drawing

The following changes occurred to the Spreadsheet Drawing schema:

- The ST_CF simple type (§19.4.3.1) was modified to allow any xsd:string
- The ST_TrueFalseBlank simple type (§20.1.2.6) now allows enumeration values of `true` and `false`

D.6 Shared

D.6.1 Shared – Bibliography

The following changes occurred to the Bibliography schema:

- The ST_String255 simple type was removed and replaced by ST_String (Part 1, §22.9.2.13)

D.6.2 Shared – Custom Properties Variant Types

The following changes occurred to the Custom Properties Variant Types schema:

- The cf element was removed
- The ST_Cf simple type was removed

D.6.3 Shared – Math

The following changes occurred to the Math schema:

- The ST_YAlign simple type (Part 1, §22.9.2.20) now uses an enumeration value of `bottom` in place of `bot`, and allows values `inside` and `outside`
- The ST_XAlign simple type (Part 1, §22.9.2.18) now allows values `inside` and `outside`

- The content model of ST_OnOff (Part 1, §22.9.2.7) was changed to an xsd:boolean, removing the values on and `off`

D.6.4 Shared Simple Types

The following changes occurred to shared simple types:

- The ST_UniversalMeasure simple type (Part 1, §22.9.2.15) was added
- The ST_AlgClass simple type (§20.1.2.1) now uses an enumeration value of `custom` in place of `invalid`
- The ST_AlgType simple type (§20.1.2.2) now uses an enumeration value of `custom` in place of `invalid`
- The ST_CryptProv simple type (§20.1.2.4) now uses an enumeration value of `custom` in place of `invalid`
- The content model of ST_OnOff (Part 1, §22.9.2.7) was changed to an xsd:boolean, removing the values on and `off`
- The content model of ST_TwipsMeasure (Part 1, §22.9.2.14) was modified to allow ST_PositiveUniversalMeasure (Part 1, §22.9.2.12)
- The ST_PositiveUniversalMeasure simple type (Part 1, §22.9.2.12) was added
- The ST_Percentage simple type (Part 1, §22.9.2.9) was added
- The ST_FixedPercentage simple type (Part 1, §22.9.2.3) was added
- The ST_PositivePercentage simple type (Part 1, §22.9.2.11) was added
- The ST_PositiveFixedPercentage simple type (Part 1, §22.9.2.10) was added

D.7 Custom XML Schema References

The following changes occurred to the Custom XML Schema References schema:

- The schemaLanguage attribute was added to the schema element (Part 1, §23.2.1)

End informative annex.

Bibliography

The following documents are useful references for implementers and users of this International Standard, in addition to the Normative References:

Information on elements, attributes, and OPC parts in ISO/IEC 29500 (OOXML), http://purl.oclc.org/ooxml/onlineInfomativeAnnexes

ICS 35.060; 35.240.30