20. 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*]

The subordinate subclauses specify the semantics for the XML markup comprising DrawingML content, which can be used within the contents of WordprocessingML, SpreadsheetML, or PresentationML documents.

This portion of DrawingML defines its core pieces.

20.1 DrawingML - Main

The DrawingML Main namespace defines all of the base constructs for all kinds of DrawingML objects (charts, diagrams, shapes, pictures, and so on). These constructs and primitives are defined below.

20.1.1 Table of Contents

This subclause is informative.

End of informative text.

20.1.2 Basics

This section describes all the basic common elements associated with the DrawingML framework.

20.1.2.1 EMU Unit of Measurement

Throughout ISO/IEC 29500, the EMU is used as a unit of measurement for length. An *EMU* is defined as follows:

$$1 \text{ emu} = \frac{1}{914400} \text{US inch} = \frac{1}{360000} \text{cm}$$

[*Rationale*: The EMU was created in order to be able to evenly divide in both English and Metric units, in order to avoid rounding errors during the calculation. The usage of EMUs also facilitates a more seamless system switch and interoperability between different locales utilizing different units of measurement. EMUs define an integer based, high precision coordinate system. *end rationale*]

20.1.2.2 Core Drawing Object Information

Within DrawingML, there is the notion of core drawing elements. These are elements that both are vital to and common across the DrawingML framework. These elements denote the most integral pieces of the DrawingML document structure and thus are among the most widely used.

[*Note*: Measurement Units - Length units must be expressed in device-independent physical units: English Metric units (EMUs), points, picas, and inches. Device-dependent units such as pixels must not be used. *end note*]

20.1.2.2.1 bldChart (Build Chart)

This element specifies how to build the animation for a diagram.

[*Example*: Consider the following example where a chart is specified to be animated by category rather than as one entity. Thus, the bldChart element should be used as follows:


```
<p:bdldLst>
  <p:bldGraphic spid="4" grpId="0">
    <p:bldSub>
      <a:bldChart bld="category"/>
    </p:bldSub>
  </p:bldGraphic>
</p:bldLst>
```

end example]

Attributes	Description
animBg (Animate Background)	Specifies whether or not the chart background elements should be animated as well. [*Note*: An example of background elements are grid lines and the chart legend. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
bld (Build)	Specifies how the chart is built. The animation animates the sub-elements in the container in the particular order defined by this attribute. The possible values for this attribute are defined by the ST_AnimationChartBuildType simple type (§20.1.10.5).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AnimationChartBuildProperties) is located in §A.4.1. *end note*]

20.1.2.2.2 bldDgm (Build Diagram)

This element specifies how to build the animation for a diagram.

[*Example*: Consider having a diagram appear as on entity as opposed to by section. The bldDgm element should be used as follows:

```
<p:bdldLst>
  <p:bldGraphic spid="4" grpId="0">
    <p:bldSub>
      <a:bldDgm bld="one"/>
    </p:bldSub>
  </p:bldGraphic>
</p:bldLst>
```

end example]

Attributes	Description
bld (Build)	Specifies how the chart is built. The animation animates the sub-elements in the container in the particular order defined by this attribute.

Attributes	Description
	The possible values for this attribute are defined by the ST_AnimationDgmBuildType simple type (§20.1.10.7).
rev (Reverse Animation)	Specifies whether the animation of the objects in this diagram should be reversed or not. If this attribute is not specified, a value of false is assumed. 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_AnimationDgmBuildProperties) is located in §A.4.1. *end note*]

20.1.2.2.3 chart (Chart to Animate)

This element specifies a reference to a chart that should be animated within a sequence of slide animations. In addition to simply acting as a reference to a chart there is also animation build steps defined.

Attributes	Description
bldStep (Animation Build Step)	Specifies which step this part of the chart should be built using. For instance the chart can be built as one object meaning it is animated as a single graphic. Alternatively the chart can be animated, or built as separate pieces. The possible values for this attribute are defined by the ST_ChartBuildStep simple type (§20.1.10.13).
categoryIdx (Category Index)	Specifies the index of the category within the corresponding chart that should be animated. The possible values for this attribute are defined by the W3C XML Schema int datatype.
seriesIdx (Series Index)	Specifies the index of the series within the corresponding chart that should be animated. 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_AnimationChartElement) is located in §A.4.1. *end note*]

20.1.2.2.4 cNvCxnSpPr (Non-Visual Connector Shape Drawing Properties)

This element specifies the non-visual drawing properties for a connector shape. These non-visual properties are properties that the generating application would utilize when rendering the slide surface.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualConnectorProperties) is located in §A.4.1. *end note*]

20.1.2.2.5 cNvGraphicFramePr (Non-Visual Graphic Frame Drawing Properties)

This element specifies the non-visual drawing properties for a graphic frame. These non-visual properties are properties that the generating application would utilize when rendering the slide surface.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualGraphicFrameProperties) is located in §A.4.1. *end note*]

20.1.2.2.6 cNvGrpSpPr (Non-Visual Group Shape Drawing Properties)

This element specifies the non-visual drawing properties for a group shape. These non-visual properties are properties that the generating application would utilize when rendering the slide surface.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualGroupDrawingShapeProps) is located in §A.4.1. *end note*]

20.1.2.2.7 cNvPicPr (Non-Visual Picture Drawing Properties)

This element specifies the non-visual properties for the picture canvas. These properties are to be used by the generating application to determine how certain properties are to be changed for the picture object in question.

[*Example*: Consider the following DrawingML.

```
<p:pic>
  …
  <p:nvPicPr>
    <p:cNvPr id="4" name="Lilly_by_Lisher.jpg"/>
    <p:cNvPicPr>
      <a:picLocks noChangeAspect="1"/>
    </p:cNvPicPr>
    <p:nvPr/>
  </p:nvPicPr>
  …
</p:pic>
```

end example]

Attributes	Description
preferRelativeResize (Relative Resize Preferred)	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

Attributes	Description
	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.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualPictureProperties) is located in §A.4.1. *end note*]

20.1.2.2.8 cNvPr (Non-Visual Drawing Properties)

This element specifies non-visual canvas properties. This allows for additional information that does not affect the appearance of the picture to be stored.

[*Example*: Consider the following DrawingML.

```
<p:pic>
  …
  <p:nvPicPr>
    <p:cNvPr id="4" name="Lilly_by_Lisher.jpg"/>
  </p:nvPicPr>
  …
</p:pic>
```

end example]

Attributes	Description
descr (Alternative Text for Object)	Specifies alternative text for the current DrawingML object, for use by assistive technologies or applications which 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 which 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)	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

Attributes	Description
	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 which 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)	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 (§20.1.10.21).
name (Name)	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)	Specifies the title (caption) of the current DrawingML object.

Attributes	Description
	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.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualDrawingProps) is located in §A.4.1. *end note*]

20.1.2.2.9 cNvSpPr (Non-Visual Shape Drawing Properties)

This element specifies the non-visual drawing properties for a shape. These properties are to be used by the generating application to determine how the shape should be dealt with

[*Example*: Consider the shape that has a shape lock applied to it.

```
<p:sp>
  <p:nvSpPr>
    <p:cNvPr id="2" name="Rectangle 1"/>
    <p:cNvSpPr>
      <a:spLocks noGrp="1"/>
    </p:cNvSpPr>
  </p:nvSpPr>
…
</p:sp>
```

This shape lock is stored within the non-visual drawing properties for this shape. *end example*]

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 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*]

Attributes	Description
	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_NonVisualDrawingShapeProps) is located in §A.4.1. *end note*]

20.1.2.2.10 cxnSp (Connection Shape)

This element specifies a connection shape that is used to connect two sp elements. Once a connection is specified using a cxnSp, it is left to the generating application to determine the exact path the connector takes. That is the connector routing algorithm is left up to the generating application as the desired path might be different depending on the specific needs of the application.

[*Example*: Consider the following connector shape that connects two regular shapes.

```
<p:spTree>
  …
  <p:sp>
    <p:nvSpPr>
      <p:cNvPr id="1" name="Rectangle 1"/>
      <p:cNvSpPr/>
      <p:nvPr/>
    </p:nvSpPr>
    …
  </p:sp>
  <p:sp>
    <p:nvSpPr>
      <p:cNvPr id="2" name="Rectangle 2"/>
      <p:cNvSpPr/>
      <p:nvPr/>
    </p:nvSpPr>
    …
  </p:sp>
  <p:cxnSp>
```



```
      <p:nvCxnSpPr>
        <p:cNvPr id="3" name="Elbow Connector 3"/>
        <p:cNvCxnSpPr>
          <a:stCxn id="1" idx="3"/>
          <a:endCxn id="2" idx="1"/>
        </p:cNvCxnSpPr>
        <p:nvPr/>
      </p:nvCxnSpPr>
      …
    </p:cxnSp>
  </p:spTree>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlConnector) is located in §A.4.1. *end note*]

20.1.2.2.11 cxnSpLocks (Connection Shape Locks)

This element specifies all locking properties for a connection shape. These properties inform the generating application about specific properties that have been previously locked and thus should not be changed.

Attributes	Description
noAdjustHandles (Disallow Showing Adjust Handles)	Specifies that the generating application should not show adjust handles for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeArrowheads (Disallow Arrowhead Changes)	Specifies that the generating application should not allow arrowhead changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeAspect (Disallow Aspect Ratio Change)	Specifies that the generating application should not allow aspect ratio changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeShapeType (Disallow Shape Type Change)	Specifies that the generating application should not allow shape type changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean

Attributes	Description
	datatype.
noEditPoints (Disallow Shape Point Editing)	Specifies that the generating application should not allow shape point changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noGrp (Disallow Shape Grouping)	Specifies that the generating application should not allow shape grouping for the corresponding connection shape. That is it cannot be combined within other shapes to form a group of shapes. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noMove (Disallow Shape Movement)	Specifies that the generating application should not allow position changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noResize (Disallow Shape Resize)	Specifies that the generating application should not allow size changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noRot (Disallow Shape Rotation)	Specifies that the generating application should not allow shape rotation changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noSelect (Disallow Shape Selection)	Specifies that the generating application should not allow selecting of the corresponding connection shape. That means also that no picture, shapes or text attached to this connection shape can be selected if this attribute has been specified. If this attribute is not specified, then a value of false is assumed. 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_ConnectorLocking) is located in §A.4.1. *end note*]

20.1.2.2.12 dgm (Diagram to Animate)

This element specifies a reference to a diagram that should be animated within a sequence of slide animations. In addition to simply acting as a reference to a diagram there is also animation build steps defined.

Attributes	Description
bldStep (Animation Build Step)	Specifies which step this part of the diagram should be built using. For instance the diagram can be built as one object meaning it is animated as a single graphic. Alternatively the diagram can be animated, or built as separate pieces. The possible values for this attribute are defined by the ST_DgmBuildStep simple type (§20.1.10.20).
id (Identifier)	Specifies the GUID of the shape for this build step in the animation. The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AnimationDgmElement) is located in §A.4.1. *end note*]

20.1.2.2.13 endCxn (Connection End)

This element specifies the ending connection that should be made by the corresponding connector shape. This connects the end tail of the connector to the final destination shape.

Attributes	Description
id (Identifier)	Specifies the id of the shape to make the final connection to. The possible values for this attribute are defined by the ST_DrawingElementId simple type (§20.1.10.21).
idx (Index)	Specifies the index into the connection site table of the final connection shape. That is there are many connection sites on a shape and it shall be specified which connection site the corresponding connector shape should connect to. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Connection) is located in §A.4.1. *end note*]

20.1.2.2.14 ext (Extension)

This element specifies an extension that is used for future extensions to the current version of DrawingML. This allows for the specifying of currently unknown elements in the future that is used for later versions of generating applications.

[*Note*: This element is not intended to reintroduce transitional schema into the strict conformance class. *end note*]

Attributes	Description
uri (Uniform Resource Identifier)	Specifies the URI, or uniform resource identifier that represents the data stored under this tag. The URI is used to identify the correct 'server' that can process the contents of this tag. The possible values for this attribute are defined by the W3C XML Schema token datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OfficeArtExtension) is located in §A.4.1. *end note*]

20.1.2.2.15 extLst (Extension List)

This element specifies the extension list within which all future extensions of element type ext is defined. The extension list along with corresponding future extensions is used to extend the storage capabilities of the DrawingML framework. This allows for various new types of data to be stored natively within the framework.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OfficeArtExtensionList) is located in §A.4.1. *end note*]

20.1.2.2.16 graphic (Graphic Object)

This element specifies the existence of a single graphic object. Document authors should refer to this element when they wish to persist a graphical object of some kind. The specification for this graphical object is provided entirely by the document author and referenced within the graphicData child element.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GraphicalObject) is located in §A.4.1. *end note*]

20.1.2.2.17 graphicData (Graphic Object Data)

This element specifies the reference to a graphic object within the document. This graphic object is provided entirely by the document authors who choose to persist this data within the document.

[*Note*: Depending on the kind of graphical object used not every generating application that supports the OOXML framework has the ability to render the graphical object. *end note*]

[*Note*: This element is not intended to reintroduce transitional schema into the strict conformance class. *end note*]

Attributes	Description
uri (Uniform Resource Identifier)	Specifies the URI, or uniform resource identifier that represents the data stored under this tag. The URI is used to identify the correct 'server' that can process the contents of

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

[*Note*: The W3C XML Schema definition of this element's content model (CT_GraphicalObjectData) is located in §A.4.1. *end note*]

20.1.2.2.18 graphicFrame (Graphic Frame)

This element specifies the existence of a graphics frame. This frame contains a graphic that was generated by an external source and needs a container in which to be displayed on the slide surface.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlGraphicalObjectFrame) is located in §A.4.1. *end note*]

20.1.2.2.19 graphicFrameLocks (Graphic Frame Locks)

This element specifies all locking properties for a graphic frame. These properties inform the generating application about specific properties that have been previously locked and thus should not be changed.

Attributes	Description
noChangeAspect (Disallow Aspect Ratio Change)	Specifies that the generating application should not allow aspect ratio changes for the corresponding graphic frame. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noDrilldown (Disallow Selection of Child Shapes)	Specifies that the generating application should not allow selecting of objects within the corresponding graphic frame but allow selecting of the graphic frame itself. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noGrp (Disallow Shape Grouping)	Specifies that the generating application should not allow shape grouping for the corresponding graphic frame. That is it cannot be combined within other shapes to form a group of shapes. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noMove (Disallow Shape Movement)	Specifies that the corresponding graphic frame cannot be moved. Objects that reside within the graphic frame can still be moved unless they also have been locked. If this attribute is not specified, then a value of false is assumed.

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noResize (Disallow Shape Resize)	Specifies that the generating application should not allow size changes for the corresponding graphic frame. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noSelect (Disallow Shape Selection)	Specifies that the generating application should not allow selecting of the corresponding picture. That means also that no picture, shapes or text attached to this picture can be selected if this attribute has been specified. If this attribute is not specified, then a value of false is assumed. [*Note*: If this attribute is specified to be true then the graphic frame cannot be selected and the objects within the graphic frame cannot be selected as well. That is the entire graphic frame including all sub-parts are considered un-selectable. *end note*] 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_GraphicalObjectFrameLocking) is located in §A.4.1. *end note*]

20.1.2.2.20 grpSp (Group shape)

This element specifies a group shape that represents many shapes grouped together. This shape is to be treated just as if it were a regular shape but instead of being described by a single geometry it is made up of all the shape geometries encompassed within it. Within a group shape each of the shapes that make up the group are specified just as they normally would. The idea behind grouping elements however is that a single transform can apply to many shapes at the same time.

[*Example*: Consider the following group shape.

```
<p:grpSp>
  <p:nvGrpSpPr>
    <p:cNvPr id="10" name="Group 9"/>
    <p:cNvGrpSpPr/>
    <p:nvPr/>
  </p:nvGrpSpPr>
```



```
<p:grpSpPr>
  <a:xfrm>
    <a:off x="838200" y="990600"/>
    <a:ext cx="2426208" cy="978408"/>
    <a:chOff x="838200" y="990600"/>
    <a:chExt cx="2426208" cy="978408"/>
  </a:xfrm>
</p:grpSpPr>
<p:sp>
…
</p:sp>
<p:sp>
…
</p:sp>
<p:sp>
…
</p:sp>
</p:grpSp>
```

In the above example we see three shapes specified within a single group. These three shapes have their position and sizes specified just as they normally would within the shape tree. The generating application should apply the transformation after the bounding box for the group shape has been calculated. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlGroupShape) is located in §A.4.1. *end note*]

20.1.2.2.21 grpSpLocks (Group Shape Locks)

This element specifies all locking properties for a connection shape. These properties inform the generating application about specific properties that have been previously locked and thus should not be changed.

Attributes	Description
noChangeAspect (Disallow Aspect Ratio Change)	Specifies that the generating application should not allow aspect ratio changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noGrp (Disallow Shape Grouping)	Specifies that the corresponding group shape cannot be grouped. That is it cannot be combined within other shapes to form a group of shapes. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noMove (Disallow	Specifies that the corresponding graphic frame cannot be moved. Objects that reside

Attributes	Description
Moving Shape)	within the graphic frame can still be moved unless they also have been locked. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noResize (Disallow Shape Resizing)	Specifies that the corresponding group shape cannot be resized. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noRot (Disallow Shape Rotation)	Specifies that the corresponding group shape cannot be rotated Objects that reside within the group can still be rotated unless they also have been locked. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noSelect (Disallow Shape Selection)	Specifies that the corresponding group shape cannot have any part of it be selected. That means that no picture, shapes or attached text can be selected either if this attribute has been specified. If this attribute is not specified, then a value of false is assumed. [*Note*: This property is inherited by sub-elements and thus all shapes within the group shape cannot be selected when this attribute is set to a value of true. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noUngrp (Disallow Shape Ungrouping)	Specifies that the generating application should not show adjust handles for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. 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_GroupLocking) is located in §A.4.1. *end note*]

20.1.2.2.22 grpSpPr (Visual Group Shape Properties)

This element specifies the properties that are to be common across all of the shapes within the corresponding group. If there are any conflicting properties within the group shape properties and the individual shape properties then the individual shape properties should take precedence.

Attributes	Description
bwMode (Black and	Specifies that the group shape should be rendered using only black and white coloring.

Attributes	Description
White Mode)	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 (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupShapeProperties) is located in §A.4.1. *end note*]

20.1.2.2.23 hlinkHover (Hyperlink for Hover)

This element specifies the hyperlink information to be activated when the user's mouse is hovered over the corresponding object. The operation of the hyperlink is to have the specified action be activated when the mouse of the user hovers over the object. When this action is activated then additional attributes can be used to specify other tasks that should be performed along with the action.

Attributes	Description
action (Action Setting)	Specifies an action that is to be taken when this hyperlink is activated. This can be used to specify a slide to be navigated to or a script of code to be run. The possible values for this attribute are defined by the W3C XML Schema string datatype.
endSnd (End Sounds)	Specifies if the URL in question should stop all sounds that are playing when it is clicked. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
highlightClick (Highlight Click)	Specifies if this attribute has already been used within this document. That is when a hyperlink has already been visited that this attribute would be utilized so the generating application can determine the color of this text. If this attribute is omitted, then a value of 0 or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
history (Add Hyperlink to Page History)	Specifies whether to add this URI to the history when navigating to it. This allows for the viewing of this presentation without the storing of history information on the viewing machine. If this attribute is omitted, then a value of 1 or true is assumed.

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Drawing Object Hyperlink Target) Namespace: http://purl.oclc.org/ooxml/officeDocument/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 (§22.8.2.1).
invalidUrl (Invalid URL)	Specifies the URL when it has been determined by the generating application that the URL is invalid. That is the generating application can still store the URL but it is known that this URL is not correct. The possible values for this attribute are defined by the W3C XML Schema string datatype.
tgtFrame (Target Frame)	Specifies the target frame that is to be used when opening this hyperlink. When the hyperlink is activated this attribute is used to determine if a new window is launched for viewing or if an existing one can be used. If this attribute is omitted, than a new window is opened. The possible values for this attribute are defined by the W3C XML Schema string datatype.
tooltip (Hyperlink Tooltip)	Specifies the tooltip that should be displayed when the hyperlink text is hovered over with the mouse. If this attribute is omitted, than the hyperlink text itself can be displayed. 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_Hyperlink) is located in §A.4.1. *end note*]

20.1.2.2.24 ln (Outline)

This element specifies an outline style that can be applied to a number of different objects such as shapes and text. The line allows for the specifying of many different types of outlines including even line dashes and bevels.

Attributes	Description
algn (Stroke Alignment)	Specifies the alignment to be used for the underline stroke. The possible values for this attribute are defined by the ST_PenAlignment simple type (§20.1.10.39).

Attributes	Description
cap (Line Ending Cap Type)	Specifies the ending caps that should be used for this line. [*Note*: Examples of cap types are rounded, flat, etc. *end note*] If this attribute is omitted, than a value of square is assumed. The possible values for this attribute are defined by the ST_LineCap simple type (§20.1.10.31).
cmpd (Compound Line Type)	Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. The possible values for this attribute are defined by the ST_CompoundLine simple type (§20.1.10.15).
w (Line Width)	Specifies the width to be used for the underline stroke. If this attribute is omitted, then a value of 0 is assumed. The possible values for this attribute are defined by the ST_LineWidth simple type (§20.1.10.35).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineProperties) is located in §A.4.1. *end note*]

20.1.2.2.25 nvCxnSpPr (Non-Visual Properties for a Connection Shape)

This element specifies all non-visual properties for a connection shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a connection shape. This allows for additional information that does not affect the appearance of the connection shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlConnectorNonVisual) is located in §A.4.1. *end note*]

20.1.2.2.26 nvGraphicFramePr (Non-Visual Properties for a Graphic Frame)

This element specifies all non-visual properties for a graphic frame. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a graphic frame. This allows for additional information that does not affect the appearance of the graphic frame to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlGraphicFrameNonVisual) is located in §A.4.1. *end note*]

20.1.2.2.27 nvGrpSpPr (Non-Visual Properties for a Group Shape)

This element specifies all non-visual properties for a group shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a group

shape. This allows for additional information that does not affect the appearance of the group shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlGroupShapeNonVisual) is located in §A.4.1. *end note*]

20.1.2.2.28 nvPicPr (Non-Visual Properties for a Picture)

This element specifies all non-visual properties for a picture. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a picture. This allows for additional information that does not affect the appearance of the picture to be stored.

[*Example*: Consider the following PresentationML.

```
<p:pic>
  …
  <p:nvPicPr>
    …
  </p:nvPicPr>
  …
</p:pic>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlPictureNonVisual) is located in §A.4.1. *end note*]

20.1.2.2.29 nvSpPr (Non-Visual Properties for a Shape)

This element specifies all non-visual properties for a shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a shape. This allows for additional information that does not affect the appearance of the shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlShapeNonVisual) is located in §A.4.1. *end note*]

20.1.2.2.30 pic (Picture)

This element specifies the existence of a picture object within the document.

[*Example*: Consider the following PresentationML that specifies the existence of a picture within a document. This picture can have non-visual properties, a picture fill as well as shape properties attached to it.


```
<p:pic>
  <p:nvPicPr>
    <p:cNvPr id="4" name="lake.JPG" descr="Picture of a Lake" />
    <p:cNvPicPr>
      <a:picLocks noChangeAspect="1"/>
    </p:cNvPicPr>
    <p:nvPr/>
  </p:nvPicPr>
  <p:blipFill>
  …
  </p:blipFill>
  <p:spPr>
  …
  </p:spPr>
</p:pic>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlPicture) is located in §A.4.1. *end note*]

20.1.2.2.31 picLocks (Picture Locks)

This element specifies all locking properties for a graphic frame. These properties inform the generating application about specific properties that have been previously locked and thus should not be changed.

Attributes	Description
noAdjustHandles (Disallow Showing Adjust Handles)	Specifies that the generating application should not show adjust handles for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeArrowheads (Disallow Arrowhead Changes)	Specifies that the generating application should not allow arrowhead changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeAspect (Disallow Aspect Ratio Change)	Specifies that the generating application should not allow aspect ratio changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

Attributes	Description
noChangeShapeType (Disallow Shape Type Change)	Specifies that the generating application should not allow shape type changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noCrop (Disallow Crop Changes)	Specifies that the generating application should not allow cropping for the corresponding picture. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noEditPoints (Disallow Shape Point Editing)	Specifies that the generating application should not allow shape point changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noGrp (Disallow Shape Grouping)	Specifies that the generating application should not allow shape grouping for the corresponding connection shape. That is it cannot be combined within other shapes to form a group of shapes. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noMove (Disallow Shape Movement)	Specifies that the generating application should not allow position changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noResize (Disallow Shape Resize)	Specifies that the generating application should not allow size changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noRot (Disallow Shape Rotation)	Specifies that the generating application should not allow shape rotation changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noSelect (Disallow Shape Selection)	Specifies that the generating application should not allow selecting of the corresponding connection shape. That means also that no picture, shapes or text attached to this connection shape can be selected if this attribute has been specified. If this attribute is

Attributes	Description
	not specified, then a value of false is assumed. 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_PictureLocking) is located in §A.4.1. *end note*]

20.1.2.2.32 snd (Hyperlink Sound)

This element specifies a sound to be played when a hyperlink within the document is activated. This sound is specified from within the parent hyperlink element.

Attributes	Description
embed (Embedded Audio File Relationship ID) Namespace: http://purl.oclc.org/ooxml/officeDocument/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 §15.2.2. *end note*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
name (Sound Name)	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.

[*Note*: The W3C XML Schema definition of this element's content model (CT_EmbeddedWAVAudioFile) is located in §A.4.1. *end note*]

20.1.2.2.33 sp (Shape)

This element specifies the existence of a single shape. A shape can either be a preset or a custom geometry, defined using the DrawingML framework. In addition to a geometry each shape can have both visual and non-visual properties attached. Text and corresponding styling information can also be attached to a shape. This shape is specified along with all other shapes within either the shape tree or group shape elements.

[*Note*: Shapes are the preferred mechanism for specifying text on a slide. *end note*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlShape) is located in §A.4.1. *end note*]

20.1.2.2.34 spLocks (Shape Locks)

This element specifies all locking properties for a shape. These properties inform the generating application about specific properties that have been previously locked and thus should not be changed.

Attributes	Description
noAdjustHandles (Disallow Showing Adjust Handles)	Specifies that the generating application should not show adjust handles for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeArrowheads (Disallow Arrowhead Changes)	Specifies that the generating application should not allow arrowhead changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeAspect (Disallow Aspect Ratio Change)	Specifies that the generating application should not allow aspect ratio changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeShapeType (Disallow Shape Type Change)	Specifies that the generating application should not allow shape type changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noEditPoints (Disallow Shape Point Editing)	Specifies that the generating application should not allow shape point changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noGrp (Disallow Shape Grouping)	Specifies that the generating application should not allow shape grouping for the corresponding connection shape. That is it cannot be combined within other shapes to form a group of shapes. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noMove (Disallow Shape Movement)	Specifies that the generating application should not allow position changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed.

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noResize (Disallow Shape Resize)	Specifies that the generating application should not allow size changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noRot (Disallow Shape Rotation)	Specifies that the generating application should not allow shape rotation changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noSelect (Disallow Shape Selection)	Specifies that the generating application should not allow selecting of the corresponding connection shape. That means also that no picture, shapes or text attached to this connection shape can be selected if this attribute has been specified. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noTextEdit (Disallow Shape Text Editing)	Specifies that the generating application should not allow editing of the shape text for the corresponding shape. If this attribute is not specified, then a value of false is assumed. 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_ShapeLocking) is located in §A.4.1. *end note*]

20.1.2.2.35 spPr (Shape Properties)

This element specifies the visual shape properties that can be applied to a shape.

Attributes	Description
bwMode (Black and White Mode)	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.

Attributes	Description
	[*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 (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeProperties) is located in §A.4.1. *end note*]

20.1.2.2.36 stCxn (Connection Start)

This element specifies the starting connection that should be made by the corresponding connector shape. This connects the head of the connector to the first shape.

Attributes	Description
id (Identifier)	Specifies the id of the shape to make the final connection to. The possible values for this attribute are defined by the ST_DrawingElementId simple type (§20.1.10.21).
idx (Index)	Specifies the index into the connection site table of the final connection shape. That is there are many connection sites on a shape and it shall be specified which connection site the corresponding connector shape should connect to. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Connection) is located in §A.4.1. *end note*]

20.1.2.2.37 style (Shape Style)

This element specifies the style information for a shape.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeStyle) is located in §A.4.1. *end note*]

20.1.2.2.38 sx (Horizontal Ratio)

This element specifies the horizontal ratio for use within a scaling calculation.

Attributes	Description
d (Denominator)	Specifies the denominator to be used within the equation.

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema long datatype.
n (Numerator)	Specifies the numerator to be used within the equation. The possible values for this attribute are defined by the W3C XML Schema long datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Ratio) is located in §A.4.1. *end note*]

20.1.2.2.39 sy (Vertical Ratio)

This element specifies the vertical ratio for use within a scaling calculation.

Attributes	Description
d (Denominator)	Specifies the denominator to be used within the equation. The possible values for this attribute are defined by the W3C XML Schema long datatype.
n (Numerator)	Specifies the numerator to be used within the equation. The possible values for this attribute are defined by the W3C XML Schema long datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Ratio) is located in §A.4.1. *end note*]

20.1.2.2.40 txBody (Shape Text Body)

This element specifies the existence of text to be contained within the corresponding shape. All visible text and visible text related properties are contained within this element. There can be multiple paragraphs and within paragraphs multiple runs of text.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBody) is located in §A.4.1. *end note*]

20.1.2.2.41 txSp (Text Shape)

This element specifies the existence of a text shape within a parent shape. This text shape is specifically used for displaying text as it has only text related child elements.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlTextShape) is located in §A.4.1. *end note*]

20.1.2.2.42 useSpRect (Use Shape Text Rectangle)

This element specifies that the text rectangle from the parent shape should be used for this text shape. If this attribute is specified then the text rectangle, or text bounding box as it is also called should have the same dimensions as the text bounding box of the parent shape within which this text shape resides.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlUseShapeRectangle) is located in §A.4.1. *end note*]

20.1.2.2.43 cpLocks (Content Part Locks)

This element specifies all locking properties for a content part. These properties inform the generating application about specific properties that have been previously locked and thus should not be changed.

Attributes	Description
noAdjustHandles (Disallow Showing Adjust Handles)	Specifies that the generating application should not show adjust handles for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeArrowheads (Disallow Arrowhead Changes)	Specifies that the generating application should not allow arrowhead changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeAspect (Disallow Aspect Ratio Change)	Specifies that the generating application should not allow aspect ratio changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeShapeType (Disallow Shape Type Change)	Specifies that the generating application should not allow shape type changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noEditPoints (Disallow Shape Point Editing)	Specifies that the generating application should not allow shape point changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noGrp (Disallow	Specifies that the generating application should not allow shape grouping for the

Attributes	Description
Shape Grouping)	corresponding connection shape. That is it cannot be combined within other shapes to form a group of shapes. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noMove (Disallow Shape Movement)	Specifies that the generating application should not allow position changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noResize (Disallow Shape Resize)	Specifies that the generating application should not allow size changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noRot (Disallow Shape Rotation)	Specifies that the generating application should not allow shape rotation changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noSelect (Disallow Shape Selection)	Specifies that the generating application should not allow selecting of the corresponding connection shape. That means also that no picture, shapes, or text attached to this connection shape can be selected if this attribute has been specified. If this attribute is not specified, then a value of false is assumed. 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_ContentPartLocking) is located in §A.4.1. *end note*]

20.1.2.3 Colors

Given its own section within DrawingML Basics, colors are an integral part of the DrawingML framework. Colors are used in virtually every object to help describe it's appearance when it is rendered on the screen. Since not every generating application wishes to represent color in the same manner, it is possible to specify color in a number of different ways.

20.1.2.3.1 alpha (Alpha)

This element specifies its input color with the specific opacity, but with its color unchanged.

Attributes	Description

val (Value)	Specifies the opacity as expressed by a percentage value.
	
	[*Example*: The following represents a green solid fill which is 50% opaque
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:alpha val="50%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.44).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositiveFixedPercentage) is located in §A.4.1. *end note*]

20.1.2.3.2 alphaMod (Alpha Modulation)

This element specifies a more or less opaque version of its input color. An alpha modulate never increases the alpha beyond 100%. A 200% alpha modulate makes a input color twice as opaque as before. A 50% alpha modulate makes a input color half as opaque as before.

Attributes	Description

val (Value)	Specifies the opacity as expressed by a percentage relative to the input color.
	
	[*Example*: The following represents a green solid fill which is 50% opaque
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:alphaMod val="50%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_PositivePercentage simple type (§20.1.10.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositivePercentage) is located in §A.4.1. *end note*]

20.1.2.3.3 alphaOff (Alpha Offset)

This element specifies a more or less opaque version of its input color. Increases or decreases the input alpha percentage by the specified percentage offset. A 10% alpha offset increases a 50% opacity to 60%. A -10% alpha offset decreases a 50% opacity to 40%. The transformed alpha values are limited to a range of 0 to 100%. A 10% alpha offset increase to a 100% opaque object still results in 100% opacity.

Attributes	Description

val (Value)	Specifies the opacity as expressed by a percentage offset increase or decrease to the input color. Increases never increase the opacity beyond 100%, decreases never decrease the opacity below 0%.
	
	[*Example*: The following represents a green solid fill which is 90% opaque
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:alphaOff val="-10%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_FixedPercentage simple type (§20.1.10.24).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FixedPercentage) is located in §A.4.1. *end note*]

20.1.2.3.4 blue (Blue)

This element specifies the input color with the specific blue component, but with the red and green color components unchanged.

Attributes	Description

val (Value)	Specifies the value of the blue component. The assigned value is specified as a percentage with 0% indicating minimal blue and 100% indicating maximum blue.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (00, FF, FF)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:blue val="100%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.5 blueMod (Blue Modulation)

This element specifies the input color with its blue component modulated by the given percentage. A 50% blue modulate reduces the blue component by half. A 200% blue modulate doubles the blue component.

Attributes	Description

val (Value)	Specifies the blue component as expressed by a percentage relative to the input color component. Increases never increase the blue component beyond 100%, decreases never decrease the blue component below 0%.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, 00, FF) to value RRGGBB= (00, 00, 80)
	
	`<a:solidFill>`
	`  <a:srgbClr val="0000FF">`
	`  <a:blueMod val="50%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.6 blueOff (Blue Offset)

This element specifies the input color with its blue component shifted, but with its red and green color components unchanged.

Attributes	Description
val (Value)	Specifies the blue component as expressed by a percentage offset increase or decrease to the input color component. Increases never increase the blue component beyond 100%, decreases never decrease the blue component below 0%.

Attributes	Description

	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, 00, FF) to value RRGGBB= (00, 00, CC)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:blueOff val="-20%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.7 comp (Complement)

This element specifies that the color rendered should be the complement of its input color with the complement being defined as such. Two colors are called complementary if, when mixed they produce a shade of grey. For instance, the complement of red which is RGB (255, 0, 0) is cyan which is RGB (0, 255, 255).

Primary colors and secondary colors are typically paired in this way:

- red and cyan (where cyan is the mixture of green and blue)
- green and magenta (where magenta is the mixture of red and blue)
- blue and yellow (where yellow is the mixture of red and green)

[*Example*:

The following represents the complement of red:

```
<a:solidFill>
  <a:srgbClr val="FF0000">
    <a:comp/>
  </a:srgbClr>
</a:solidFill>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ComplementTransform) is located in §A.4.1. *end note*]

20.1.2.3.8 gamma (Gamma)

This element specifies that the output color rendered by the generating application should be the sRGB gamma shift of the input color.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GammaTransform) is located in §A.4.1. *end note*]

20.1.2.3.9 gray (Gray)

This element specifies a grayscale of its input color, taking into relative intensities of the red, green, and blue primaries.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GrayscaleTransform) is located in §A.4.1. *end note*]

20.1.2.3.10 green (Green)

This elements specifies the input color with the specified green component, but with its red and blue color components unchanged.

Attributes	Description

val (Value)	Specifies the value of the green component. The assigned value is specified as a percentage with 0% indicating minimal green and 100% indicating maximum green.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, 00, FF) to value RRGGBB= (00, FF, FF)
	
	`<a:solidFill>`
	`  <a:srgbClr val="0000FF">`
	`  <a:green val="100%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.11 greenMod (Green Modulation)

This element specifies the input color with its green component modulated by the given percentage. A 50% green modulate reduces the green component by half. A 200% green modulate doubles the green component.

Attributes	Description

val (Value)	Specifies the green component as expressed by a percentage relative to the input color component. Increases never increase the green component beyond 100%, decreases never decrease the green component below 0%.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (00, 80, 00)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:greenMod val="50%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.12 greenOff (Green Offset)

This element specifies the input color with its green component shifted, but with its red and blue color components unchanged.

Attributes	Description

val (Value)	Specifies the green component as expressed by a percentage offset increase or decrease to the input color component. Increases never increase the green component beyond 100%, decreases never decrease the green component below 0%.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (00, CC, 00)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:greenOff val="-20%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.13 hslClr (Hue, Saturation, Luminance Color Model)

This element specifies a color using the HSL color model. A perceptual gamma of 2.2 is assumed.

Hue refers to the dominant wavelength of color, saturation refers to the purity of its hue, and luminance refers to its lightness or darkness.

As with all colors, colors defined with the HSL color model can have color transforms applied to it.

[*Example*:

The color blue having RGB value RRGGBB = (00, 00, 80) is equivalent to

```
<a:solidFill>
  <a:hslClr hue="14400000" sat="100%" lum="50%">
</a:solidFill>
```

end example]

Attributes	Description
hue (Hue)	Specifies the angular value describing the wavelength. Expressed in 1/6000ths of a degree. The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.43).
lum (Luminance)	Specifies the luminance referring to the lightness or darkness of the color. Expressed as a percentage with 0% referring to maximal dark (black) and 100% referring to maximal white. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
sat (Saturation)	Specifies the saturation referring to the purity of the hue. Expressed as a percentage with 0% referring to grey, 100% referring to the purest form of the hue. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_HslColor) is located in §A.4.1. *end note*]

20.1.2.3.14 hue (Hue)

This element specifies the input color with the specified hue, but with its saturation and luminance unchanged.

[*Example*: The following two solid fills are equivalent.

```
<a:solidFill>
  <a:hslClr hue="14400000" sat="100%" lum="50%">
</a:solidFill>
<a:solidFill>
  <a:hslClr hue="0" sat="100%" lum="50%">
    <a:hue val="14400000"/>
  <a:hslClr/>
</a:solidFill>
```

end example]

Attributes	Description
val (Value)	Specifies the actual angle value to be used with the input color's hue component. The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.43).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositiveFixedAngle) is located in §A.4.1. *end note*]

20.1.2.3.15 hueMod (Hue Modulate)

This element specifies the input color with its hue modulated by the given percentage. A 50% hue modulate decreases the angular hue value by half. A 200% hue modulate doubles the angular hue value.

Attributes	Description

val (Value)	Specifies the opacity as expressed by a percentage relative to the input color.
	
	[*Example*: The following represents a green solid fill which is 50% opaque
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:alphaMod val="50%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_PositivePercentage simple type (§20.1.10.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositivePercentage) is located in §A.4.1. *end note*]

20.1.2.3.16 hueOff (Hue Offset)

This element specifies the input color with its hue shifted, but with its saturation and luminance unchanged.

Attributes	Description

val (Value)	Specifies the actual angular value of the shift. The result of the shift shall be between 0 and 360 degrees. Shifts resulting in angular values less than 0 are treated as 0. Shifts resulting in angular values greater than 360 are treated as 360.
	
	[*Example*:
	The following increases the hue angular value by 10 degrees.
	
	`<a:solidFill>`
	`  <a:hslClr hue="0" sat="100%" lum="50%"/>`
	`  <a:hueOff val="600000"/>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Angle simple type (§20.1.10.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Angle) is located in §A.4.1. *end note*]

20.1.2.3.17 inv (Inverse)

This element specifies the inverse of its input color.

[*Example*:

The inverse of red (1, 0, 0) is cyan (0, 1, 1).

The following represents cyan, the inverse of red:

```
<a:solidFill>
  <a:srgbClr val="FF0000">
    <a:inv/>
  </a:srgbClr>
</a:solidFill>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_InverseTransform) is located in §A.4.1. *end note*]

20.1.2.3.18 invGamma (Inverse Gamma)

This element specifies that the output color rendered by the generating application should be the inverse sRGB gamma shift of the input color.

[*Note*: The W3C XML Schema definition of this element's content model (CT_InverseGammaTransform) is located in §A.4.1. *end note*]

20.1.2.3.19 lum (Luminance)

This element specifies the input color with the specified luminance, but with its hue and saturation unchanged. Typically luminance values fall in the range [0%, 100%].

[*Example*:

The following two solid fills are equivalent:

```
<a:solidFill>
  <a:hslClr hue="14400000" sat="100%" lum="50%">
</a:solidFill>
<a:solidFill>
  <a:hslClr hue="14400000" sat="100%" lum="0%">
    <a:lum val="50%"/>
  <a:hslClr/>
</a:solidFill>
```

end example]

Attributes	Description

val (Value)	Specifies the value of the luminance. The assigned value is specified as a percentage with 0% indicating minimal luminance and 100% indicating maximum luminance.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (00, 66, 00)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:lum val="20%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.20 lumMod (Luminance Modulation)

This element specifies the input color with its luminance modulated by the given percentage. A 50% luminance modulate reduces the luminance by half. A 200% luminance modulate doubles the luminance.

Attributes	Description

val (Value)	Specifies the luminance as expressed by a percentage relative to the input color. Increases never increase the luminance beyond 100%, decreases never decrease the luminance below 0%.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (00, 75, 00)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:lumMod val="50%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.21 lumOff (Luminance Offset)

This element specifies the input color with its luminance shifted, but with its hue and saturation unchanged.

Attributes	Description

val (Value)	Specifies the luminance as expressed by a percentage offset increase or decrease to the input color. Increases never increase the luminance beyond 100%, decreases never decrease the luminance below 0%.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (00, 99, 00)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:lumOff val="-20%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.22 prstClr (Preset Color)

This element specifies a color which is bound to one of a predefined collection of colors.

[*Example*:

The following defines a solid fill bound to the "black" preset color.

```
<a:solidFill>
  <a:prstClr val="black">
</a:solidFill>
```

end example]

Attributes	Description
val (Value)	Specifies the actual preset color value. The possible values for this attribute are defined by the ST_PresetColorVal simple type (§20.1.10.47).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PresetColor) is located in §A.4.1. *end note*]

20.1.2.3.23 red (Red)

This element specifies the input color with the specified red component, but with its green and blue color components unchanged.

Attributes	Description
val (Value)	Specifies the value of the red component. The assigned value is specified as a percentage with 0% indicating minimal red and 100% indicating maximum red. [*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (FF, FF, 00)

Attributes	Description
	<a:solidFill> <a:srgbClr val="00FF00"> <a:red val="100%"/> </a:srgbClr> </a:solidFill> *end example*] The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.24 redMod (Red Modulation)

This element specifies the input color with its red component modulated by the given percentage. A 50% red modulate reduces the red component by half. A 200% red modulate doubles the red component.

Attributes	Description
val (Value)	Specifies the red component as expressed by a percentage relative to the input color component. Increases never increase the red component beyond 100%, decreases never decrease the red component below 0%. [*Example*: The following manipulates the fill from having RGB value RRGGBB = (FF, 00, 00) to value RRGGBB= (80, 00, 00) <a:solidFill> <a:srgbClr val="FF0000"> <a:redMod val="50%"/> </a:srgbClr> </a:solidFill> *end example*] The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.25 redOff (Red Offset)

This element specifies the input color with its red component shifted, but with its green and blue color components unchanged.

Attributes	Description

val (Value)	Specifies the red component as expressed by a percentage offset increase or decrease to the input color component. Increases never increase the red component beyond 100%, decreases never decrease the red component below 0%.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (FF, 00, 00) to value RRGGBB= (CC, 00, 00)
	
	`<a:solidFill>`
	`  <a:srgbClr val="FF0000">`
	`  <a:redOff val="-20%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.26 sat (Saturation)

This element specifies the input color with the specified saturation, but with its hue and luminance unchanged. Typically saturation values fall in the range [0%, 100%].

[*Example*:

The following two solid fills are equivalent:

```
<a:solidFill>
  <a:hslClr hue="14400000" sat="100%" lum="50%">
</a:solidFill>
<a:solidFill>
  <a:hslClr hue="14400000" sat="0%" lum="50%">
    <a:sat val="100000"/>
  <a:hslClr/>
</a:solidFill>
```

end example]

Attributes	Description
val (Value)	Specifies the value of the saturation. The assigned value is specified as a percentage with 0% indicating minimal saturation and 100% indicating maximum saturation.

Attributes	Description

	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (40, C0, 40)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:sat val="50%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.27 satMod (Saturation Modulation)

This element specifies the input color with its saturation modulated by the given percentage. A 50% saturation modulate reduces the saturation by half. A 200% saturation modulate doubles the saturation.

Attributes	Description

val (Value)	Specifies the saturation as expressed by a percentage relative to the input color. Increases never increase the saturation beyond 100%, decreases never decrease the saturation below 0%.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (66, 99, 66)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:satMod val="20%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.28 satOff (Saturation Offset)

This element specifies the input color with its saturation shifted, but with its hue and luminance unchanged. A 10% offset to 20% saturation yields 30% saturation.

Attributes	Description

val (Value)	Specifies the saturation as expressed by a percentage offset increase or decrease to the input color. Increases never increase the saturation beyond 100%, decreases never decrease the saturation below 0%.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (19, E5, 19)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:satOff val="-20%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.29 schemeClr (Scheme Color)

This element specifies a color bound to a user's theme. As with all elements which define a color, it is possible to apply a list of color transforms to the base color defined.

Attributes	Description

val (Value)	Specifies the desired scheme.
	
	[*Example*: The following represents a color bound to the "lt1" theme color
	
	`<a:solidFill>`
	`  <a:schemeClr val="lt1"/>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_SchemeColorVal simple type (§20.1.10.53).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SchemeColor) is located in §A.4.1. *end note*]

20.1.2.3.30 scrgbClr (RGB Color Model - Percentage Variant)

This element specifies a color using the red, green, blue RGB color model. Each component, red, green, and blue is expressed as a percentage from 0% to 100%. A linear gamma of 1.0 is assumed.

Specifies the level of red as expressed by a percentage offset increase or decrease relative to the input color.

[*Example*: The following represent the same color

```
<a:solidFill>
  <a:scrgbClr r="50%" g="50%" b="50%"/>
</a:solidFill>
<a:solidFill>
  <a:srgbClr val="BCBCBC"/>
</a:solidFill>
```

end example]

Attributes	Description
b (Blue)	Specifies the percentage of blue. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
g (Green)	Specifies the percentage of green. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
r (Red)	Specifies the percentage of red. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ScRgbColor) is located in §A.4.1. *end note*]

20.1.2.3.31 shade (Shade)

This element specifies a darker version of its input color. A 10% shade is 10% of the input color combined with 90% black.

Attributes	Description
val (Value)	Specifies the shade as expressed by a percentage value.

Attributes	Description

	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (00, BC, 00)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:shade val="50%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.44).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositiveFixedPercentage) is located in §A.4.1. *end note*]

20.1.2.3.32 srgbClr (RGB Color Model - Hex Variant)

This element specifies a color using the red, green, blue RGB color model. Red, green, and blue is expressed as sequence of hex digits, RRGGBB. A perceptual gamma of 2.2 is used.

Specifies the level of red as expressed by a percentage offset increase or decrease relative to the input color.

[*Example*: The following represent the same color

```
<a:solidFill>
  <a:scrgbClr r="50%" g="50%" b="50%"/>
</a:solidFill>
<a:solidFill>
  <a:srgbClr val="BCBCBC"/>
</a:solidFill>
```

end example]

Attributes	Description
val (Value)	The actual color value. Expressed as a sequence of hex digits RRGGBB. The possible values for this attribute are defined by the ST_HexColorRGB simple type (§22.9.2.5).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SRgbColor) is located in §A.4.1. *end note*]

20.1.2.3.33 sysClr (System Color)

This element specifies a color bound to predefined operating system elements.

[*Example*: The following represents the default color used for displaying text in a window.

```
<a:solidFill>
  <a:sysClr val="windowText"/>
</a:solidFill>
```

end example]

Attributes	Description
lastClr (Last Color)	Specifies the color value that was last computed by the generating application. The possible values for this attribute are defined by the ST_HexColorRGB simple type (§22.9.2.5).
val (Value)	Specifies the system color value. The possible values for this attribute are defined by the ST_SystemColorVal simple type (§20.1.10.57).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SystemColor) is located in §A.4.1. *end note*]

20.1.2.3.34 tint (Tint)

This element specifies a lighter version of its input color. A 10% tint is 10% of the input color combined with 90% white.

Attributes	Description

val (Value)	Specifies the tint as expressed by a percentage value.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (BC, FF, BC)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:tint val="50%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.44).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositiveFixedPercentage) is located in §A.4.1. *end note*]

20.1.3 Audio and Video

The Audio and Video portion of the DrawingML framework deals with all media of these two kinds that can be attached to objects within a document. Types of audio that can be represented within a file are CD audio, QuickTime audio, and any other generic audio. When dealing with generic audio there is the option for embedding it within the file and also linking it. The linking option is preferable if the size of the audio file is too large and thus increases the size of the document by an undesirable amount. For video there are two kinds that can be represented and that is either a QuickTime movie or any other generic movie. When dealing with generic video there is only the option of linking to the media as video is too large to embed within a document.

20.1.3.1 audioCd (Audio from CD)

This element specifies the existence of Audio from a CD. This element is specified within the non-visual properties of an object. The audio shall be attached to an object as this is how it is represented within the document. The actual playing of the sound however is done within the timing node list that is specified under the timing element.

[*Example*: Consider the following picture object that has an audio from a CD attached to it.

```
<p:pic>
  <p:nvPicPr>
    <p:cNvPr id="7" name="Rectangle 6">
      <a:hlinkClick r:id="" action="ppaction://media"/>
    </p:cNvPr>
    <p:cNvPicPr>
      <a:picLocks noRot="1"/>
    </p:cNvPicPr>
    <p:nvPr>
      <a:audioCd>
        <a:st track="1"/>
        <a:end track="3" time="65"/>
      </a:audioCd>
    </p:nvPr>
  </p:nvPicPr>
  …
</p:pic>
```

In the above example, we see that there is a single audioCD element attached to this picture. This picture is placed within the document just as a normal picture or shape would be. The id of this picture, namely 7 in this case, is used to refer to this audioCD element from within the timing node list. For this example we see that the audio for this CD starts playing at the 0 second mark on the first track and ends on the 1 minute 5 second mark of the third track. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_AudioCD) is located in §A.4.1. *end note*]

20.1.3.2 audioFile (Audio from File)

This element specifies the existence of an audio file. This element is specified within the non-visual properties of an object. The audio shall be attached to an object as this is how it is represented within the document. The actual playing of the audio however is done within the timing node list that is specified under the timing element.

[*Example*: Consider the following picture object that has an audio file attached to it.

```
<p:pic>
  <p:nvPicPr>
    <p:cNvPr id="7" name="Rectangle 6">
      <a:hlinkClick r:id="" action="ppaction://media"/>
    </p:cNvPr>
    <p:cNvPicPr>
      <a:picLocks noRot="1"/>
    </p:cNvPicPr>
    <p:nvPr>
      <a:audioFile r:link="rId1"/>
    </p:nvPr>
  </p:nvPicPr>
  …
</p:pic>
```

In the above example, we see that there is a single audioFile element attached to this picture. This picture is placed within the document just as a normal picture or shape would be. The id of this picture, namely 7 in this case, is used to refer to this audioFile element from within the timing node list. The Linked relationship id is used to retrieve the actual audio file for playback purposes. *end example*]

Attributes	Description
contentType (Content Type of Linked Audio File)	Specifies the content type for the external file that is referenced by this element. Content types define a media type, a subtype, and an optional set of parameters, as defined in Part 2. If a rendering application cannot process external content of the content type specified, then the specified content can be ignored. [*Note*: A list of suggested audio types is provided in §15.2.2. *end note*] If this attribute is omitted, application should attempt to determine the content type by reading the contents of the relationship's target. A producer that wants interoperability should use the following standard format: • audio/mpeg ISO/IEC 11172-3 The possible values for this attribute are defined by the W3C XML Schema string

Attributes	Description
	datatype.
link (Linked Relationship ID) Namespace: http://purl.oclc.org/ooxml/officeDocument/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 (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AudioFile) is located in §A.4.1. *end note*]

20.1.3.3 end (Audio End Time)

This element specifies the end point for a CD Audio sound element. Encompassed within this element are the time and track at which the sound should halt its playback. This element is used in conjunction with an Audio Start Time element to specify the time span for an entire audioCD sound element.

[*Example*: Consider the following DrawingML.

```
<a:audioCd>
  <a:st track="1" time="2"/>
  <a:end track="3" time="65"/>
</a:audioCd>
```

In the above example, the audioCD sound element shown specifies for a portion of audio spanning from 2 seconds into the first track to 1 minute, 5 seconds into the third track. *end example*]

Attributes	Description
time (Time)	Specifies the time in seconds that the CD Audio should be started at. If this attribute is omitted, then a value of 0 is assumed. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
track (Track)	Specifies which track of the CD this Audio begins playing on. This attribute is required and cannot be omitted. The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AudioCDTime) is located in §A.4.1. *end note*]

20.1.3.4 quickTimeFile (QuickTime from File)

This element specifies the existence of a QuickTime file, as defined in the 2007-09-04 version of the QuickTime File Format Specification: http://developer.apple.com/documentation/QuickTime/QTFF/qtff.pdf. [*Note*: For more information on the QuickTime format: http://developer.apple.com/reference/QuickTime/. *end note*]. This element is specified within the non-visual properties of an object. The QuickTime file shall be attached to an object as this is how it is represented within the document. The actual playing of the QuickTime however is done within the timing node list that is specified under the timing element.

[*Example*: Consider the following picture object that has a QuickTime file attached to it.

```
<p:pic>
  <p:nvPicPr>
    <p:cNvPr id="7" name="Rectangle 6">
      <a:hlinkClick r:id="" action="ppaction://media"/>
    </p:cNvPr>
    <p:cNvPicPr>
      <a:picLocks noRot="1"/>
    </p:cNvPicPr>
    <p:nvPr>
      <a:quickTimeFile r:link="rId1"/>
    </p:nvPr>
  </p:nvPicPr>
  …
</p:pic>
```

In the above example, we see that there is a single quickTimeFile element attached to this picture. This picture is placed within the document just as a normal picture or shape would be. The id of this picture, namely 7 in this case, is used to refer to this quickTimeFile element from within the timing node list. The Linked relationship id is used to retrieve the actual video file for playback purposes. *end example*]

Attributes	Description
link (Linked Relationship ID) Namespace: http://purl.oclc.org/ooxml/officeDocument/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 (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_QuickTimeFile) is located in §A.4.1. *end note*]

20.1.3.5 st (Audio Start Time)

This element specifies the start point for a CD Audio sound element. Encompassed within this element are the time and track at which the sound should begin its playback. This element is used in conjunction with an Audio End Time element to specify the time span for an entire audioCD sound element.

[*Example*: Consider the following DrawingML.

```
<a:audioCd>
  <a:st track="1" time="2"/>
  <a:end track="3" time="65"/>
</a:audioCd>
```

In the above example, the audioCD sound element shown specifies for a portion of audio spanning from 2 seconds into the first track to 1 minute, 5 seconds into the third track. *end example*]

Attributes	Description
time (Time)	Specifies the time in seconds that the CD Audio should be started at. If this attribute is omitted, then a value of 0 is assumed. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
track (Track)	Specifies which track of the CD this Audio begins playing on. This attribute is required and cannot be omitted. The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AudioCDTime) is located in §A.4.1. *end note*]

20.1.3.6 videoFile (Video from File)

This element specifies the existence of a video file. This element is specified within the non-visual properties of an object. The video shall be attached to an object as this is how it is represented within the document. The actual playing of the video however is done within the timing node list that is specified under the timing element.

[*Example*: Consider the following picture object that has a video attached to it.

```
<p:pic>
  <p:nvPicPr>
    <p:cNvPr id="7" name="Rectangle 6">
      <a:hlinkClick r:id="" action="ppaction://media"/>
    </p:cNvPr>
```

```
      <p:cNvPicPr>
        <a:picLocks noRot="1"/>
      </p:cNvPicPr>
      <p:nvPr>
        <a:videoFile r:link="rId1"/>
      </p:nvPr>
    </p:nvPicPr>
    …
  </p:pic>
```

In the above example, we see that there is a single videoFile element attached to this picture. This picture is placed within the document just as a normal picture or shape would be. The id of this picture, namely 7 in this case, is used to refer to this videoFile element from within the timing node list. The Linked relationship id is used to retrieve the actual video file for playback purposes. *end example*]

Attributes	Description
contentType (Content Type of Linked Video File)	Specifies the content type for the external file that is referenced by this element. Content types define a media type, a subtype, and an optional set of parameters, as defined in Part 2. If a rendering application cannot process external content of the content type specified, then the specified content can be ignored. [*Note*: A list of suggested video types is provided in §15.2.17. *end note*] If this attribute is omitted, application should attempt to determine the content type by reading the contents of the relationship's target. The possible values for this attribute are defined by the W3C XML Schema string datatype.
link (Linked Relationship ID) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the identification information for a linked video file. 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 (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_VideoFile) is located in §A.4.1. *end note*]

20.1.3.7 wavAudioFile (Audio from WAV File)

This element specifies the existence of an audio WAV file. This element is specified within the non-visual properties of an object. The audio shall be attached to an object as this is how it is represented within the document. The actual playing of the audio however is done within the timing node list that is specified under the timing element.

[*Example*: Consider the following picture object that has an audio WAV file attached to it.

```
<p:pic>
  <p:nvPicPr>
    <p:cNvPr id="7" name="Rectangle 6">
      <a:hlinkClick r:id="" action="ppaction://media"/>
    </p:cNvPr>
    <p:cNvPicPr>
      <a:picLocks noRot="1"/>
    </p:cNvPicPr>
    <p:nvPr>
      <a:wavAudioFile r:embed="rId2"/>
    </p:nvPr>
  </p:nvPicPr>
  …
</p:pic>
```

In the above example, we see that there is a single wavAudioFile element attached to this picture. This picture is placed within the document just as a normal picture or shape would be. The id of this picture, namely 7 in this case, is used to refer to this wavAudioFile element from within the timing node list. The Embedded relationship id is used to retrieve the actual audio file for playback purposes. *end example*]

[*Note*: This element is generally used for the purposes of embedding audio files within the document. For linking to generic audio files the audioFile element should be used. *end note*]

Attributes	Description
embed (Embedded Audio File Relationship ID) Namespace: http://purl.oclc.org/ooxml/officeDocument/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 §15.2.2. *end note*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
name (Sound Name)	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.

[*Note*: The W3C XML Schema definition of this element's content model (CT_EmbeddedWAVAudioFile) is located in §A.4.1. *end note*]

20.1.4 Styles

Styles within DrawingML refer to the way a particular object (be it text or a shape, or anything else) is formatted. Different aspects, ranging from color, line type, fill, and effects applied to the object can be predefined within a theme. The main purpose of a theme is to define a style matrix from which a document can pull style information from in order to format the visual look of objects in a document.

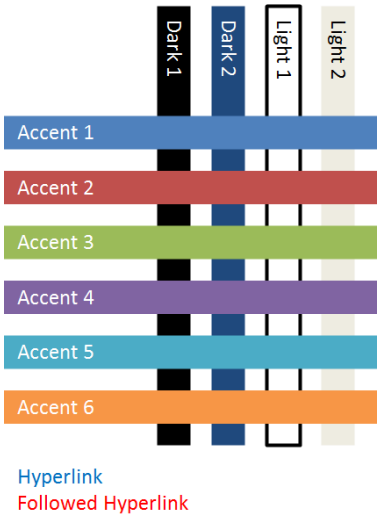
20.1.4.1 Styles

The elements in this section compose the basic definition of a style, including its associated colors, effect styles, line styles, fill styles, background styles, and font scheme.

20.1.4.1.1 accent1 (Accent 1)

This element defines a color that happens to be the accent 1 color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:



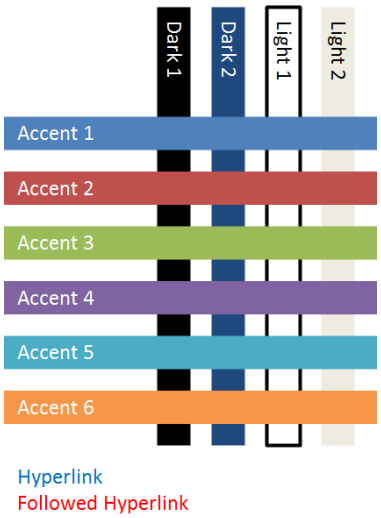
end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.4.1.2 accent2 (Accent 2)

This element defines a color that happens to be the accent 2 color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:



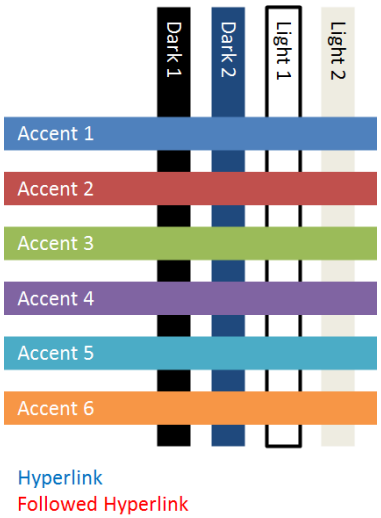
end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.4.1.3 accent3 (Accent 3)

This element defines a color that happens to be the accent 3 color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:



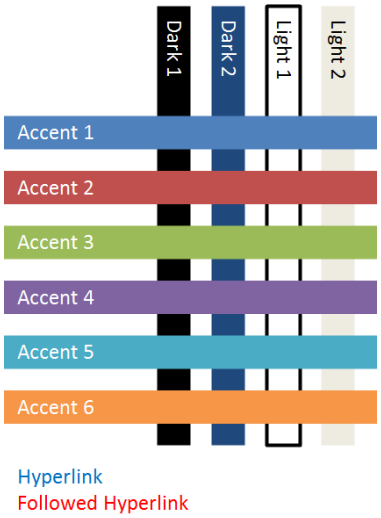
end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.4.1.4 accent4 (Accent 4)

This element defines a color that happens to be the accent 4 color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:



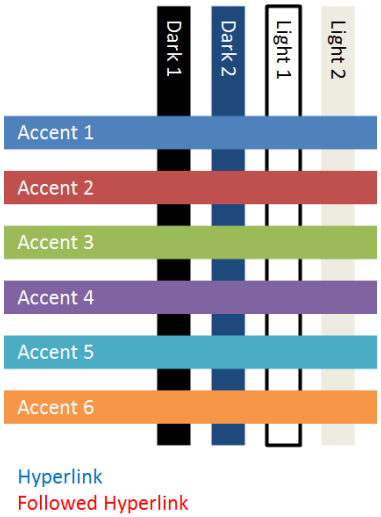
end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.4.1.5 accent5 (Accent 5)

This element defines a color that happens to be the accent 5 color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:



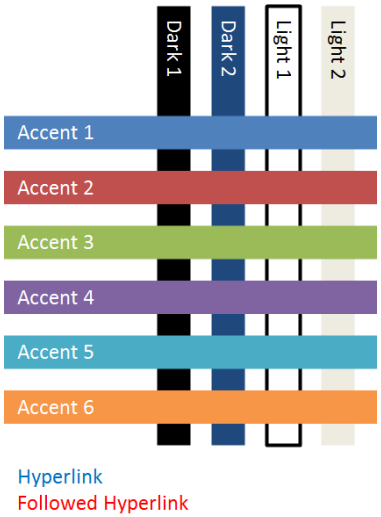
end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.4.1.6 accent6 (Accent 6)

This element defines a color that happens to be the accent 1 color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:



end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.4.1.7 bgFillStyleLst (Background Fill Style List)

This element defines a list of background fills that are used within a theme. The background fills consist of three fills, arranged in order from subtle to moderate to intense.

[*Example*: Consider the following example of a background fill style list within DrawingML:

```
<bgFillStyleLst>
  <solidFill>
…
  </solidFill>
  <gradFill rotWithShape="1">
…
  </gradFill>
  <blipFill>
…
  </blipFill>
</bgFillStyleLst>
```

In this example, we see that the list contains a solid fill for the subtle fill, a gradient fill for the moderate fill and an image fill for the intense background fill. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_BackgroundFillStyleList) is located in §A.4.1. *end note*]

20.1.4.1.8 custClr (Custom color)

This element defines a custom color. The custom colors are used within a custom color list to define custom colors that are extra colors that can be appended to a theme. This is useful within corporate scenarios where there is a set corporate color palette from which to work.

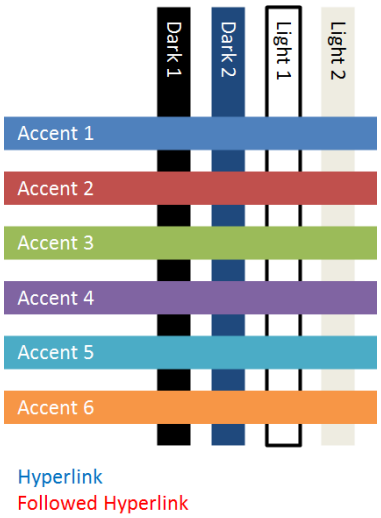
Attributes	Description
name (Name)	The name of the color shown in the color picker. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomColor) is located in §A.4.1. *end note*]

20.1.4.1.9 dk1 (Dark 1)

This element defines a color that happens to be the dark 1 color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:



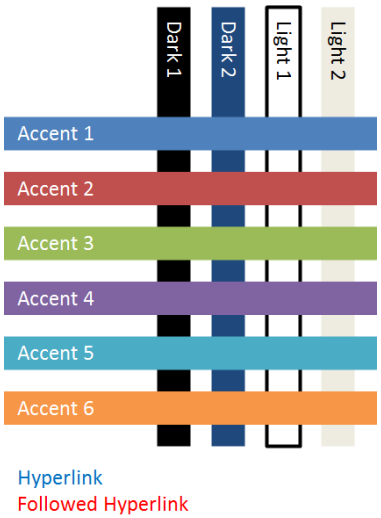
end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.4.1.10 dk2 (Dark 2)

This element defines a color that happens to be the dark 2 color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:



end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.4.1.11 effectStyle (Effect Style)

This element defines a set of effects and 3D properties that can be applied to an object.

[*Example*: Consider the following example of an effect style within DrawingML:

```
<effectStyle>
  <effectLst>
    <outerShdw blurRad="57150" dist="38100" dir="5400000" algn="ctr"
      rotWithShape="0">
      <schemeClr val="phClr">
        <shade val="9000"/>
        <satMod val="105000"/>
        <alpha val="48000"/>
      </schemeClr>
    </outerShdw>
  </effectLst>
</effectStyle>
```

In this example, an outer shadow is being applied to a shape as the moderate effect. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_EffectStyleItem) is located in §A.4.1. *end note*]

20.1.4.1.12 effectStyleLst (Effect Style List)

This element defines a set of three effect styles that create the effect style list for a theme. The effect styles are arranged in order of subtle to moderate to intense.

[*Example*: Consider the following example of an effect style list within DrawingML:

```
<effectStyleLst>
  <effectStyle>
    <effectLst>
      <outerShdw blurRad="57150" dist="38100" dir="5400000"
        algn="ctr" rotWithShape="0">
…
      </outerShdw>
    </effectLst>
  </effectStyle>
```



```
    <effectStyle>
      <effectLst>
        <outerShdw blurRad="57150" dist="38100" dir="5400000"
          algn="ctr" rotWithShape="0">
…
        </outerShdw>
      </effectLst>
    </effectStyle>
    <effectStyle>
      <effectLst>
        <outerShdw blurRad="57150" dist="38100" dir="5400000"
          algn="ctr" rotWithShape="0">
…
        </outerShdw>
      </effectLst>
      <scene3d>
…
      </scene3d>
      <sp3d prstMaterial="powder">
…
      </sp3d>
    </effectStyle>
  </effectStyleLst>
```

In this example, we see three effect styles defined. The first two (subtle and moderate) define an outer shadow as the effect, while the third effect style (intense) defines an outer shadow along with 3D properties which are to be applied to the object as well. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_EffectStyleList) is located in §A.4.1. *end note*]

20.1.4.1.13 fillStyleLst (Fill Style List)

This element defines a set of three fill styles that are used within a theme. The three fill styles are arranged in order from subtle to moderate to intense.

[*Example*: Consider the following example of a fill style list within DrawingML:

```
  <fillStyleLst>
    <solidFill>
…
    </solidFill>
    <gradFill rotWithShape="1">
…
    </gradFill>
```



```
    <gradFill rotWithShape="1">
  …
    </gradFill>
  </fillStyleLst>
```

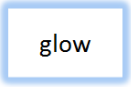
In this example, we see three fill styles being defined within the fill style list. The first style is the subtle style and defines simply a solid fill. The second and third styles (moderate and intense fills respectively) define gradient fills. *end example*]

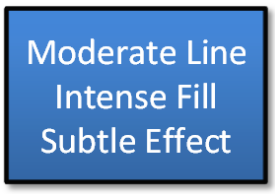
[*Note*: The W3C XML Schema definition of this element's content model (CT_FillStyleList) is located in §A.4.1. *end note*]

20.1.4.1.14 fmtScheme (Format Scheme)

This element contains the background fill styles, effect styles, fill styles, and line styles which define the style matrix for a theme. The style matrix consists of subtle, moderate, and intense fills, lines, and effects. The background fills are not generally thought of to directly be associated with the matrix, but do play a role in the style of the overall document. Usually, a given object chooses a single line style, a single fill style, and a single effect style in order to define the overall final look of the object.

[*Example*: Consider the following example of the style matrix in use within DrawingML:

	Line	Fill	Effect
Subtle			
Moderate			
Intense			



In this example, we see a shape styled which utilizes different aspects from the above defined style matrix. *end example*]

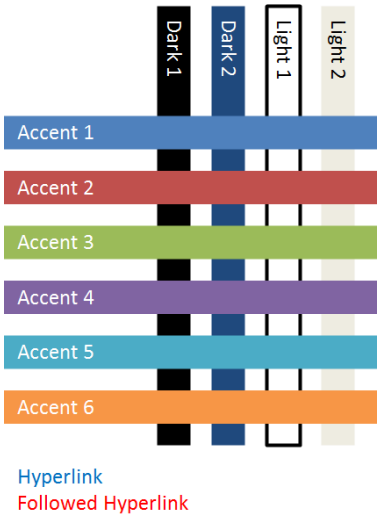
Attributes	Description
name (Name)	Defines the name for the format scheme. The name is simply a human readable string which identifies the format scheme in the user interface. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StyleMatrix) is located in §A.4.1. *end note*]

20.1.4.1.15 folHlink (Followed Hyperlink)

This element defines a color that happens to be the followed hyperlink color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:



end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.4.1.16 font (Font)

This element defines a font within the styles area of DrawingML. A font is defined by a script along with a typeface.

[*Example*: Consider the following example of a font in DrawingML:

```
<font script="Thai" typeface="Cordia New"/>
```

In this example, we see that the script 'Thai' is supposed to use the font face 'Cordia New'. *end example*]

Attributes	Description
script (Script)	Specifies the script, or language, in which the typeface is supposed to be used. [*Note*: It is recommended that script names as specified in ISO 15924 are used. *end note*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
typeface (Typeface)	Specifies the font face to use. The possible values for this attribute are defined by the ST_TextTypeface simple type (§20.1.10.80).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SupplementalFont) is located in §A.4.1. *end note*]

20.1.4.1.17 fontRef (Font Reference)

This element represents a reference to a themed font. When used it specifies which themed font to use along with a choice of color.

[*Example*: Consider the following example of a font reference within DrawingML:

```
<fontRef idx="minor">
  <schemeClr val="tx1"/>
</fontRef>
```

In this example, we see a font referencing the minor font defined within the theme. *end example*]

Attributes	Description
idx (Identifier)	Specifies the identifier of the font to reference. The possible values for this attribute are defined by the ST_FontCollectionIndex simple type (§20.1.10.25).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontReference) is located in §A.4.1. *end note*]

20.1.4.1.18 fontScheme (Font Scheme)

This element defines the font scheme within the theme. The font scheme consists of a pair of major and minor fonts for which to use in a document. The major font corresponds well with the heading areas of a document, and the minor font corresponds well with the normal text or paragraph areas.

[*Example*: Consider the following example of a font scheme within DrawingML:


```
<fontScheme name="sample">
  <majorFont>
…
  </majorFont>
  <minorFont>
…
  </minorFont>
</fontScheme>
```

In this example, we see the major and minor font lists within the font scheme that is named 'sample'. *end example*]

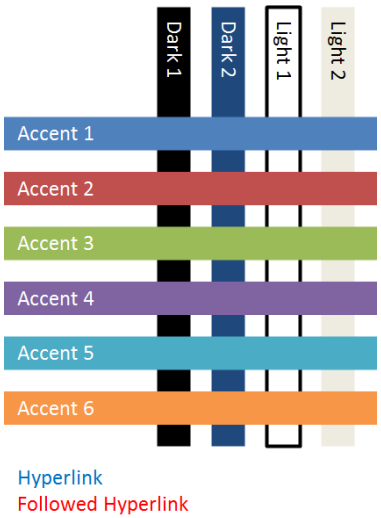
Attributes	Description
name (Name)	The name of the font scheme shown in the user interface. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontScheme) is located in §A.4.1. *end note*]

20.1.4.1.19 hlink (Hyperlink)

This element defines a color that happens to be the hyperlink color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:



end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.4.1.20 lnDef (Line Default)

This element defines a default line that is used within a document.

[*Example*: Consider the following example of a default line defined in DrawingML:

```
<lnDef>
  <spPr/>
  <bodyPr/>
  <lstStyle/>
  <style>
    <lnRef idx="1">
      <schemeClr val="accent2"/>
    </lnRef>
    <fillRef idx="0">
      <schemeClr val="accent2"/>
    </fillRef>
    <effectRef idx="0">
      <schemeClr val="accent2"/>
    </effectRef>
    <fontRef idx="minor">
      <schemeClr val="tx1"/>
    </fontRef>
  </style>
</lnDef>
```

In this example, we see that the default line for the document is being defined as a themed line which references the subtle line style with idx equal to 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DefaultShapeDefinition) is located in §A.4.1. *end note*]

20.1.4.1.21 lnStyleLst (Line Style List)

This element defines a list of three line styles for use within a theme. The three line styles are arranged in order from subtle to moderate to intense versions of lines. This list makes up part of the style matrix.

[*Example*: Consider the following example of a line style list within DrawingML:


```
<lnStyleLst>
  <ln w="9525" cap="flat" cmpd="sng" algn="ctr">
    <solidFill>
      <schemeClr val="phClr">
        <shade val="50000"/>
        <satMod val="103000"/>
      </schemeClr>
    </solidFill>
    <prstDash val="solid"/>
  </ln>
  <ln w="25400" cap="flat" cmpd="sng" algn="ctr">
    <solidFill>
      <schemeClr val="phClr"/>
    </solidFill>
    <prstDash val="solid"/>
  </ln>
  <ln w="38100" cap="flat" cmpd="sng" algn="ctr">
    <solidFill>
      <schemeClr val="phClr"/>
    </solidFill>
    <prstDash val="solid"/>
  </ln>
</lnStyleLst>
```

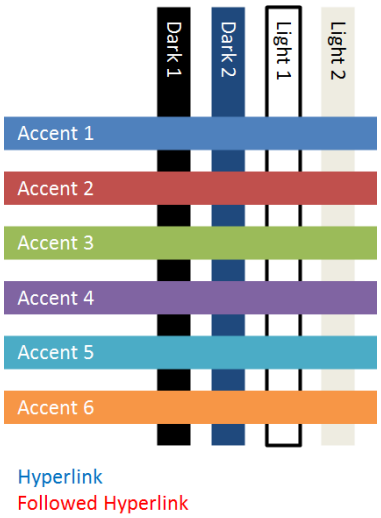
In this example, we see three lines defined within a line style list. The first line corresponds to the subtle line, the second to the moderate, and the third corresponds to the intense line defined in the theme. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineStyleList) is located in §A.4.1. *end note*]

20.1.4.1.22 lt1 (Light 1)

This element defines a color that happens to be the accent 1 color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:



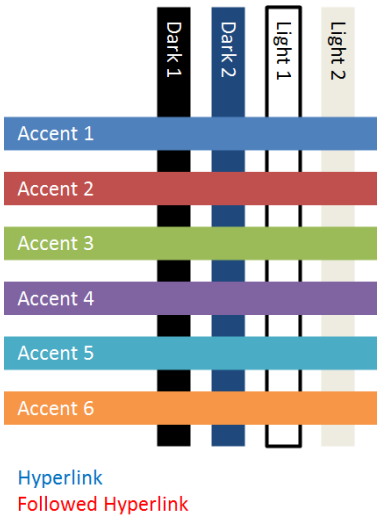
end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.4.1.23 lt2 (Light 2)

This element defines a color that happens to be the accent 1 color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:



end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.4.1.24 majorFont (Major Font)

This element defines the set of major fonts which are to be used under different languages or locals.

[*Example*: Consider the following example of the major fonts being defined within DrawingML:

```
<majorFont>
  <latin typeface="Calibri"/>
  <ea typeface="Arial"/>
  <cs typeface="Arial"/>
  <font script="Jpan" typeface="ＭＳ Ｐゴシック"/>
  <font script="Hang" typeface="HY중고딕"/>
  <font script="Hans" typeface="隶书"/>
  <font script="Hant" typeface="微軟正黑體"/>
  <font script="Arab" typeface="Traditional Arabic"/>
  <font script="Hebr" typeface="Arial"/>
  <font script="Thai" typeface="Cordia New"/>
  <font script="Ethi" typeface="Nyala"/>
  <font script="Beng" typeface="Vrinda"/>
  <font script="Gujr" typeface="Shruti"/>
  <font script="Khmr" typeface="DaunPenh"/>
  <font script="Knda" typeface="Tunga"/>
</majorFont>
```

In this example, we see the latin, east asian, and complex script fonts defined along with many fonts for different locals. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontCollection) is located in §A.4.1. *end note*]

20.1.4.1.25 minorFont (Minor fonts)

This element defines the set of minor fonts that are to be used under different languages or locals.

[*Example*: Consider the following example of the minor fonts being defined within DrawingML:

```
<minorFont>
  <latin typeface="Calibri"/>
  <ea typeface="Arial"/>
  <cs typeface="Arial"/>
  <font script="Jpan" typeface="ＭＳ Ｐゴシック"/>
  <font script="Hang" typeface="HY중고딕"/>
  <font script="Hans" typeface="隶书"/>
  <font script="Hant" typeface="微軟正黑體"/>
  <font script="Arab" typeface="Traditional Arabic"/>
  <font script="Hebr" typeface="Arial"/>
  <font script="Thai" typeface="Cordia New"/>
  <font script="Ethi" typeface="Nyala"/>
  <font script="Beng" typeface="Vrinda"/>
  <font script="Gujr" typeface="Shruti"/>
  <font script="Khmr" typeface="DaunPenh"/>
  <font script="Knda" typeface="Tunga"/>
</minorFont>
```

In this example, we see the latin, east asian, and complex script fonts defined along with many fonts for different locals. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontCollection) is located in §A.4.1. *end note*]

20.1.4.1.26 scene3d (3D Scene Properties)

This element defines optional scene-level 3D properties to apply to an object.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Scene3D) is located in §A.4.1. *end note*]

20.1.4.1.27 spDef (Shape Default)

This element defines the formatting that is associated with the default shape. The default formatting can be applied to a shape when it is initially inserted into a document.

[*Example*: Consider the following example of a shape default being used within DrawingML:

```
<spDef>
  <spPr>
    <solidFill>
      <schemeClr val="accent2">
        <shade val="75000"/>
      </schemeClr>
    </solidFill>
  </spPr>
```



```
    <bodyPr rtlCol="0" anchor="ctr"/>
    <lstStyle>
      <defPPr algn="ctr">
        <defRPr/>
      </defPPr>
    </lstStyle>
    <style>
      <lnRef idx="1">
        <schemeClr val="accent1"/>
      </lnRef>
      <fillRef idx="2">
        <schemeClr val="accent1"/>
      </fillRef>
      <effectRef idx="1">
        <schemeClr val="accent1"/>
      </effectRef>
      <fontRef idx="minor">
        <schemeClr val="dk1"/>
      </fontRef>
    </style>
  </spDef>
```

In this example, we see a default shape which references a certain themed fill, line, effect, and font along with an override fill to these. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DefaultShapeDefinition) is located in §A.4.1. *end note*]

20.1.4.1.28 txDef (Text Default)

This element defines the default formatting which is applied to text in a document by default. The default formatting can and should be applied to the shape when it is initially inserted into a document.

[*Example*: Consider the following example of a text default being used within DrawingML:

```
  <txDef>
    <spPr>
      <solidFill>
        <schemeClr val="accent2">
          <shade val="75000"/>
        </schemeClr>
      </solidFill>
    </spPr>
    <bodyPr rtlCol="0" anchor="ctr"/>
```

```
    <lstStyle>
      <defPPr algn="ctr">
        <defRPr/>
      </defPPr>
    </lstStyle>
    <style>
      <lnRef idx="1">
        <schemeClr val="accent1"/>
      </lnRef>
      <fillRef idx="2">
        <schemeClr val="accent1"/>
      </fillRef>
      <effectRef idx="1">
        <schemeClr val="accent1"/>
      </effectRef>
      <fontRef idx="minor">
        <schemeClr val="dk1"/>
      </fontRef>
    </style>
  </txDef>
```

In this example, we see a default text which references a certain themed fill, line, effect, and font along with an override fill to these. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DefaultShapeDefinition) is located in §A.4.1. *end note*]

20.1.4.2 Table Styles

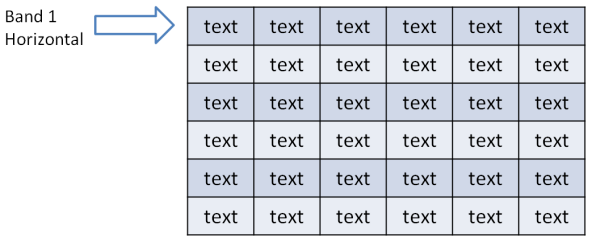
Table styles are responsible for the rapid formatting that can be applied to a table. This rapid formatting takes different parts of a table into account, such as if the first row or last row should be emphasized, or if there is some type of banding (row for example) present on the table. All of these different types of formatting can be defined within a table style

20.1.4.2.1 band1H (Band 1 Horizontal)

This element describes the formatting for the first row in horizontal banding. Two different row formatting are applied to the table alternating in order to create a banding effect on the table.

[*Example*: Consider the following example of band 1 horizontal being used within DrawingML:


```
<band1H>
  <tcStyle>
    <tcBdr/>
    <fill>
      <solidFill>
        <schemeClr val="accent1">
          <tint val="40000"/>
        </schemeClr>
      </solidFill>
    </fill>
  </tcStyle>
</band1H>
```



In this example, we set the fill to be a solid fill referencing the accent 1 color defined in the theme. *end example*]

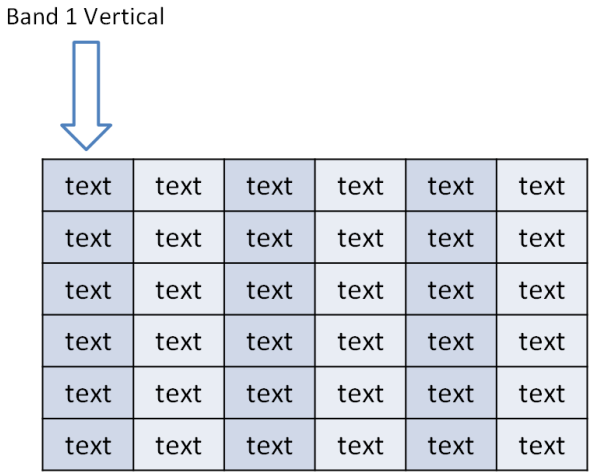
[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.2 band1V (Band 1 Vertical)

This element describes the formatting for the first row in vertical banding. Two different row formatting are applied to the table alternating in order to create a banding effect on the table.

[*Example*: Consider the following example of band 1 vertical being used within DrawingML:

```
<band1V>
  <tcStyle>
    <tcBdr/>
    <fill>
      <solidFill>
        <schemeClr val="accent1">
          <tint val="40000"/>
        </schemeClr>
      </solidFill>
    </fill>
  </tcStyle>
</band1V>
```



In this example, we set the fill to be a solid fill referencing the accent 1 color defined in the theme. *end example*]

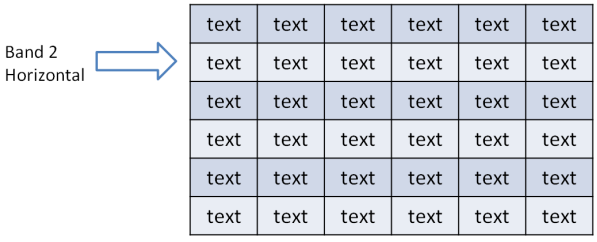
[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.3 band2H (Band 2 Horizontal)

This element describes the formatting for the second row in horizontal banding. Two different row formatting are applied to the table alternating in order to create a banding effect on the table.

[*Example*: Consider the following example of band 2 horizontal being used within DrawingML:

```
<band2H>
  <tcStyle>
    <tcBdr/>
    <fill>
      <solidFill>
        <schemeClr val="accent2">
          <tint val="40000"/>
        </schemeClr>
      </solidFill>
    </fill>
  </tcStyle>
</band2H>
```



In this example, we set the fill to be a solid fill referencing the accent 2 color defined in the theme. *end example*]

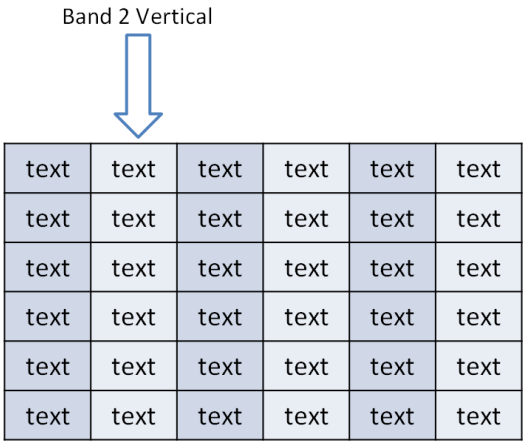
[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.4 band2V (Band 2 Vertical)

This element describes the formatting for the second row in vertical banding. Two different row formatting are applied to the table alternating in order to create a banding effect on the table.

[*Example*: Consider the following example of band 2 vertical being used within DrawingML:

```
<band2V>
  <tcStyle>
    <tcBdr/>
    <fill>
      <solidFill>
        <schemeClr val="accent2">
          <tint val="40000"/>
        </schemeClr>
      </solidFill>
    </fill>
  </tcStyle>
</band2V>
```

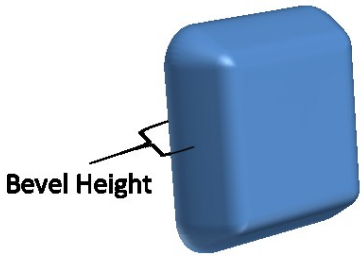
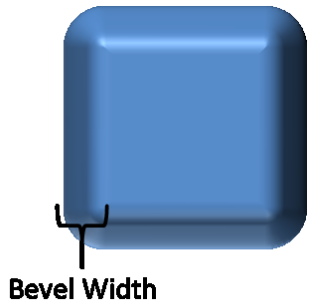


In this example, we set the fill to be a solid fill referencing the accent 2 color defined in the theme. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.5 bevel (Bevel)

This element defines the properties of the bevel associated with the 3D effect applied to a cell in a table.

Attributes	Description
h (Height)	Specifies the height of the bevel, or how far above the shape it is applied. [*Example*: Consider the following example bevel  In this example, we see the height of an example bevel on a shape. *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
prst (Preset Bevel)	Specifies the preset bevel type which defines the look of the bevel. The possible values for this attribute are defined by the ST_BevelPresetType simple type (§20.1.10.9).
w (Width)	Specifies the width of the bevel, or how far into the shape it is applied. [*Example*: Consider the following example bevel  In this example, we see the width of an example bevel on a shape. *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Bevel) is located in §A.4.1. *end note*]

20.1.4.2.6 bottom (Bottom Border)

This element defines the line properties associated with the bottom border in a table cell.

[*Example*: Consider the following example of the bottom border in use within DrawingML:

```
<bottom>
  <ln w="12700" cmpd="sng">
    <solidFill>
      <schemeClr val="accent1"/>
    </solidFill>
  </ln>
</bottom>
```

In this example, we see the bottom border on a table cell to be a single 1pt line which is colored accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ThemeableLineStyle) is located in §A.4.1. *end note*]

20.1.4.2.7 effect (Effect)

This element defines the effect that can be applied to a table as a whole through a table style.

[*Example*: Consider the following example of an effect in use within DrawingML:

```
<effect>
  <effectLst>
    <glow rad="228600">
      <schemeClr val="accent1">
        <satMod val="175000"/>
        <alpha val="40000"/>
      </schemeClr>
    </glow>
  </effectLst>
</effect>
```

In this example, we see a glow being defined within the table style that is applied to the table as a whole. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_EffectProperties) is located in §A.4.1. *end note*]

20.1.4.2.8 effectRef (Effect Reference)

This element defines a reference to an effect style within the style matrix. The idx attribute refers the index of an effect style within the effectStyleLst element.

Attributes	Description
idx (Style Matrix Index)	Specifies the style matrix index of the style referred to. The possible values for this attribute are defined by the ST_StyleMatrixColumnIndex simple type (§20.1.10.56).

[*Note*: The W3C XML Schema definition of this element's content model (CT_StyleMatrixReference) is located in §A.4.1. *end note*]

20.1.4.2.9 fill (Fill)

This element defines the fill that is applied to the table as a whole. The background of the table can contain a single fill that is the entire size of the table. This can allow for gradient fills, or image fills, which span the entire size of the table.

[*Example*: Consider the following example of a fill on a table background in DrawingML:

```
<fill>
  <gradFill flip="none" rotWithShape="1">
    <gsLst>
      <gs pos="0">
        <schemeClr val="accent2">
          <shade val="75000"/>
        </schemeClr>
      </gs>
      <gs pos="100000">
        <schemeClr val="accent2">
          <shade val="75000"/>
          <tint val="20000"/>
        </schemeClr>
      </gs>
    </gsLst>
    <lin ang="2700000" scaled="1"/>
    <tileRect/>
  </gradFill>
</fill>
```

In this example, we apply a gradient fill to the entire table on the background shape of the table. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_FillProperties) is located in §A.4.1. *end note*]

20.1.4.2.10 fillRef (Fill Reference)

This element defines a reference to a fill style within the style matrix. The idx attribute refers to the index of a fill style or background fill style within the presentation's style matrix, defined by the fmtScheme element. A value of 0 or 1000 indicates no background, values 1-999 refer to the index of a fill style within the fillStyleLst element, and values 1001 and above refer to the index of a background fill style within the bgFillStyleLst element. The value 1001 corresponds to the first background fill style, 1002 to the second background fill style, and so on.

[*Example*:

```
<a:fillRef idx="2">
  <a:schemeClr val="accent2"/>
</a:fillRef>
```

The above code indicates the object is to have the style's second fill style using the accent2 color of the color scheme.

end example]

[*Example*:

```
<a:fillRef idx="1001">
  <a:schemeClr val="accent2"/>
</a:fillRef>
```

The above code indicates the object is to have the style's first background fill style using the accent2 color of the color scheme.

end example]

Attributes	Description
idx (Style Matrix Index)	Specifies the style matrix index of the style referred to. The possible values for this attribute are defined by the ST_StyleMatrixColumnIndex simple type (§20.1.10.56).

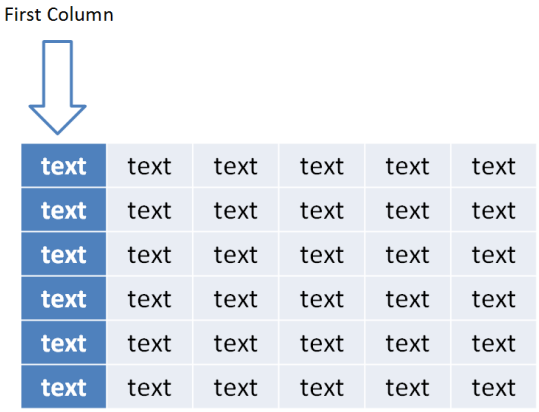
[*Note*: The W3C XML Schema definition of this element's content model (CT_StyleMatrixReference) is located in §A.4.1. *end note*]

20.1.4.2.11 firstCol (First Column)

This element defines the cell formatting which can be applied to the first column of the table.

[*Example*: Consider the following example of first column formatting within DrawingML:

```
<firstCol>
  <tcTxStyle b="on">
    <fontRef idx="minor">
      <scrgbClr r="0" g="0" b="0"/>
    </fontRef>
    <schemeClr val="lt1"/>
  </tcTxStyle>
  <tcStyle>
    <tcBdr/>
    <fill>
      <solidFill>
        <schemeClr val="accent1"/>
      </solidFill>
    </fill>
  </tcStyle>
</firstCol>
```



In this example, we define the first column cell fills to be accent 1 along with the text properties to be bold when first column formatting is enabled through the user interface. *end example*]

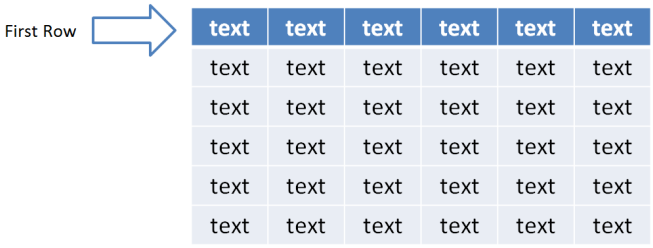
[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.12 firstRow (First Row)

This element defines the cell formatting which can be applied to the first row of the table.

[*Example*: Consider the following example of first row formatting within DrawingML:


```
<firstRow>
  <tcTxStyle b="on">
    <fontRef idx="minor">
      <scrgbClr r="0" g="0" b="0"/>
    </fontRef>
    <schemeClr val="lt1"/>
  </tcTxStyle>
  <tcStyle>
    <tcBdr/>
    <fill>
      <solidFill>
        <schemeClr val="accent1"/>
      </solidFill>
    </fill>
  </tcStyle>
</firstRow>
```



In this example, we define the first row cell fills to be accent 1 along with the text properties to be bold when first row formatting is enabled through the user interface. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.13 font (Font)

This element defines the font to be used within a given table cell text style. This element allows for exact definition of the font within the table style instead of referencing a themed font.

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontCollection) is located in §A.4.1. *end note*]

20.1.4.2.14 insideH (Inside Horizontal Border)

This element defines the line properties associated with the inner horizontal borders in a table.

[*Example*: Consider the following example of the inner horizontal borders in use within DrawingML:

```
<insideH>
  <ln w="12700" cmpd="sng">
    <solidFill>
      <schemeClr val="accent1"/>
    </solidFill>
  </ln>
</insideH>
```

In this example, we see the inner horizontal borders in a table to be a single 1pt line which is colored accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ThemeableLineStyle) is located in §A.4.1. *end note*]

20.1.4.2.15 insideV (Inside Vertical Border)

This element defines the line properties associated with the inner vertical borders in a table.

[*Example*: Consider the following example of the inside vertical borders in use within DrawingML:

```
<insideV>
  <ln w="12700" cmpd="sng">
    <solidFill>
      <schemeClr val="accent1"/>
    </solidFill>
  </ln>
</insideV>
```

In this example, we see the inner vertical borders in a table to be a single 1pt line which is colored accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ThemeableLineStyle) is located in §A.4.1. *end note*]

20.1.4.2.16 lastCol (Last Column)

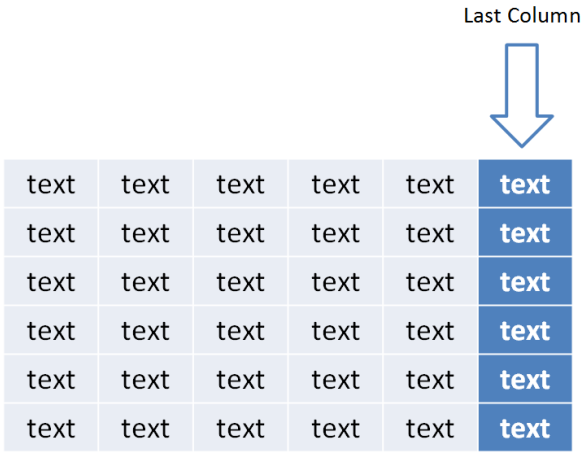
This element defines the cell formatting which can be applied to the last column of the table.

[*Example*: Consider the following example of last column formatting within DrawingML:

```
<lastCol>
  <tcTxStyle b="on">
    <fontRef idx="minor">
      <scrgbClr r="0" g="0" b="0"/>
    </fontRef>
    <schemeClr val="lt1"/>
  </tcTxStyle>
```



```
    <tcStyle>
      <tcBdr/>
      <fill>
        <solidFill>
          <schemeClr val="accent1"/>
        </solidFill>
      </fill>
    </tcStyle>
  </lastCol>
```



In this example, we define the last column cell fills to be accent 1 along with the text properties to be bold when last column formatting is enabled through the user interface. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.17 lastRow (Last Row)

This element defines the cell formatting which can be applied to the last row of the table.

[*Example*: Consider the following example of last row formatting within DrawingML:

```
  <lastRow>
    <tcTxStyle b="on">
      <fontRef idx="minor">
        <scrgbClr r="0" g="0" b="0"/>
      </fontRef>
      <schemeClr val="lt1"/>
    </tcTxStyle>
```

```
    <tcStyle>
      <tcBdr/>
      <fill>
        <solidFill>
          <schemeClr val="accent1"/>
        </solidFill>
      </fill>
    </tcStyle>
  </lastRow>
```

In this example, we define the last row cell fills to be accent 1 along with the text properties to be bold when last row formatting is enabled through the user interface. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.18 left (Left Border)

This element defines the line properties associated with the left border in a table cell.

[*Example*: Consider the following example of the left border in use within DrawingML:

```
  <left>
    <ln w="12700" cmpd="sng">
      <solidFill>
        <schemeClr val="accent1"/>
      </solidFill>
    </ln>
  </left>
```

In this example, we see the left border on a table cell to be a single 1pt line which is colored accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ThemeableLineStyle) is located in §A.4.1. *end note*]

20.1.4.2.19 lnRef (Line Reference)

This element defines a reference to a line style within the style matrix. The idx attribute refers the index of a line style within the fillStyleLst element.

Attributes	Description
idx (Style Matrix Index)	Specifies the style matrix index of the style referred to. The possible values for this attribute are defined by the ST_StyleMatrixColumnIndex simple type (§20.1.10.56).

[*Note*: The W3C XML Schema definition of this element's content model (CT_StyleMatrixReference) is located in §A.4.1. *end note*]

20.1.4.2.20 neCell (Northeast Cell)

This element defies the formatting for the cell in the northeast corner of a table when both the first row formatting and last column formatting are enabled. This formatting is only applied to the single cell which overlaps between the two formatting options.

[*Example*: Consider the following example of the northeast cell formatting within DrawingML:

```
<neCell>
  <tcTxStyle b="on">
    <fontRef idx="minor">
      <scrgbClr r="0" g="0" b="0"/>
    </fontRef>
    <schemeClr val="lt1"/>
  </tcTxStyle>
  <tcStyle>
    <tcBdr/>
    <fill>
      <solidFill>
        <schemeClr val="accent1"/>
      </solidFill>
    </fill>
  </tcStyle>
</neCell>
```

In this example, we specifically set the northeast cell to contain bold text with a solid cell fill in the color of accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.21 nwCell (Northwest Cell)

This element defies the formatting for the cell in the northwest corner of a table when both the first row formatting and first column formatting are enabled. This formatting is only applied to the single cell which overlaps between the two formatting options.

[*Example*: Consider the following example of the northwest cell formatting within DrawingML:

```
<nwCell>
  <tcTxStyle b="on">
    <fontRef idx="minor">
      <scrgbClr r="0" g="0" b="0"/>
    </fontRef>
    <schemeClr val="lt1"/>
  </tcTxStyle>
  <tcStyle>
    <tcBdr/>
    <fill>
      <solidFill>
        <schemeClr val="accent1"/>
      </solidFill>
    </fill>
  </tcStyle>
</nwCell>
```

In this example, we specifically set the northwest cell to contain bold text with a solid cell fill in the color of accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.22 right (Right Border)

This element defines the line properties associated with the right border in a table cell.

[*Example*: Consider the following example of the right border in use within DrawingML:

```
<right>
  <ln w="12700" cmpd="sng">
    <solidFill>
      <schemeClr val="accent1"/>
    </solidFill>
  </ln>
</right>
```

In this example, we see the right border on a table cell to be a single 1pt line which is colored accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ThemeableLineStyle) is located in §A.4.1. *end note*]

20.1.4.2.23 seCell (Southeast Cell)

This element defies the formatting for the cell in the southeast corner of a table when both the last row formatting and last column formatting are enabled. This formatting is only applied to the single cell which overlaps between the two formatting options.

[*Example*: Consider the following example of the southeast cell formatting within DrawingML:

```
<seCell>
  <tcTxStyle b="on">
    <fontRef idx="minor">
      <scrgbClr r="0" g="0" b="0"/>
    </fontRef>
    <schemeClr val="lt1"/>
  </tcTxStyle>
  <tcStyle>
    <tcBdr/>
    <fill>
      <solidFill>
        <schemeClr val="accent1"/>
      </solidFill>
    </fill>
  </tcStyle>
</seCell>
```

In this example, we specifically set the southeast cell to contain bold text with a solid cell fill in the color of accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.24 swCell (Southwest Cell)

This element defies the formatting for the cell in the southwest corner of a table when both the last row formatting and first column formatting are enabled. This formatting is only applied to the single cell which overlaps between the two formatting options.

[*Example*: Consider the following example of the southwest cell formatting within DrawingML:

```
<swCell>
  <tcTxStyle b="on">
    <fontRef idx="minor">
      <scrgbClr r="0" g="0" b="0"/>
    </fontRef>
    <schemeClr val="lt1"/>
  </tcTxStyle>
  <tcStyle>
    <tcBdr/>
    <fill>
      <solidFill>
        <schemeClr val="accent1"/>
      </solidFill>
    </fill>
  </tcStyle>
</swCell>
```

In this example, we specifically set the southwest cell to contain bold text with a solid cell fill in the color of accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.25 tblBg (Table Background)

This element defines the formatting options which can be applied to the table background shape. The background shape is the same size as the entire table and can hold a fill or an effect which spans the entire table.

[*Example*: Consider the following example of a table background in use within DrawingML:

```
<tblBg>
  <fillRef idx="2">
    <schemeClr val="accent1"/>
  </fillRef>
  <effectRef idx="1">
    <schemeClr val="accent1"/>
  </effectRef>
</tblBg>
```

In this example, we see that there is a themed fill and themed effect being applied to the table background through the table style. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableBackgroundStyle) is located in §A.4.1. *end note*]

20.1.4.2.26 tblStyle (Table Style)

This is the root element for a table style. Within the table style are different formatting options available in order to apply a table.

Attributes	Description
styleId (Style ID)	Specifies a GUID identifying the table style in a unique manner. The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).
styleName (Name)	Specifies the name of the table style which can show up in the user interface identifying the style to a user. 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_TableStyle) is located in §A.4.1. *end note*]

20.1.4.2.27 tblStyleLst (Table Style List)

This element is simply a list of table styles which are used within a document.

[*Example*: Consider the following example of a table style list within DrawingML:

```
<tblStyleLst def="{5C22544A-7EE6-4342-B048-85BDC9FD1C3A}">
  <tblStyle styleId="{5C22544A-7EE6-4342-B048-85BDC9FD1C3A}"
    styleName="Medium Style 2 - Accent 1">
    …
  </tblStyle>
  <tblStyle styleId="{3C2FFA5D-87B4-456A-9821-1D502468CF0F}"
    styleName="Themed Style 1 - Accent 1">
    …
  </tblStyle>
</tblStyleLst>
```

In this example, we see two table styles defined along with the default being specified. *end example*]

Attributes	Description
def (Default)	The GUID corresponding to the default table style in the list of table styles. This default can be used when a table is initially inserted into a document. The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableStyleList) is located in §A.4.1. *end note*]

20.1.4.2.28 tcBdr (Table Cell Borders)

This element defines the borders for the cells within a table.

[*Example*: Consider the following example of table cell borders being used within DrawingML:

```
<tcBdr>
  <left>
    <lnRef idx="1">
      <schemeClr val="accent1"/>
    </lnRef>
  </left>
  <right>
    <lnRef idx="1">
      <schemeClr val="accent1"/>
    </lnRef>
  </right>
  <top>
    <lnRef idx="1">
      <schemeClr val="accent1"/>
    </lnRef>
  </top>
  <bottom>
    <lnRef idx="2">
      <schemeClr val="lt1"/>
    </lnRef>
  </bottom>
  <insideH>
    <ln>
      <noFill/>
    </ln>
  </insideH>
  <insideV>
    <ln>
      <noFill/>
    </ln>
  </insideV>
</tcBdr>
```

In this example, we define borders for the bottom, top, right, and left borders of the table cells. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableCellBorderStyle) is located in §A.4.1. *end note*]

20.1.4.2.29 tcStyle (Table Cell Style)

This element defines the style for a give cell in a table.

[*Example*: Consider the following example of a table cell style in use within DrawingML:

```
<tcStyle>
  <tcBdr>
    …
  </tcBdr>
  <fill>
    …
  </fill>
</tcStyle>
```

In this example, we see that a set of borders for the cell along with a cell fill are being defined. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableStyleCellStyle) is located in §A.4.1. *end note*]

20.1.4.2.30 tcTxStyle (Table Cell Text Style)

This element defines the text properties associated with the text contained within a table cell.

[*Example*: Consider the following example of a table cell text style in use within DrawingML:

```
<tcTxStyle b="on">
  <fontRef idx="minor">
    <scrgbClr r="0" g="0" b="0"/>
  </fontRef>
  <schemeClr val="lt1"/>
</tcTxStyle>
```

In this example, we define the text within the cell to be bold and reference the themed minor font and to also be the light 1 color. *end example*]

Attributes	Description
b (Bold)	Specifies if the text is to be bolded. The possible values for this attribute are defined by the ST_OnOffStyleType simple type (§20.1.10.36).
i (Italic)	Specifies if the text is to be italicized. The possible values for this attribute are defined by the ST_OnOffStyleType simple type

Attributes	Description
	(§20.1.10.36).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableStyleTextStyle) is located in §A.4.1. *end note*]

20.1.4.2.31 tl2br (Top Left to Bottom Right Border)

This element defines the line properties associated with the border which goes from the top-left to the bottom-right corner in a table cell.

[*Example*: Consider the following example of the top border in use within DrawingML:

```
<tl2br>
  <ln w="12700" cmpd="sng">
    <solidFill>
      <schemeClr val="accent1"/>
    </solidFill>
  </ln>
</tl2br>
```

In this example, we see the border on a table cell to be a single 1pt line which is colored accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ThemeableLineStyle) is located in §A.4.1. *end note*]

20.1.4.2.32 top (Top Border)

This element defines the line properties associated with the top border in a table cell.

[*Example*: Consider the following example of the top border in use within DrawingML:

```
<top>
  <ln w="12700" cmpd="sng">
    <solidFill>
      <schemeClr val="accent1"/>
    </solidFill>
  </ln>
</top>
```

In this example, we see the top border on a table cell to be a single 1pt line which is colored accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ThemeableLineStyle) is located in §A.4.1. *end note*]

20.1.4.2.33 tr2bl (Top Right to Bottom Left Border)

This element defines the line properties associated with the border which goes from the top-right to the bottom-left corner in a table cell.

[*Example*: Consider the following example of the top border in use within DrawingML:

```
<tr2bl>
  <ln w="12700" cmpd="sng">
    <solidFill>
      <schemeClr val="accent1"/>
    </solidFill>
  </ln>
</tr2bl>
```

In this example, we see the border on a table cell to be a single 1pt line which is colored accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ThemeableLineStyle) is located in §A.4.1. *end note*]

20.1.4.2.34 wholeTbl (Whole Table)

This element contains formatting options which are applied to the table as a whole when it is in its default state with no formatting options (first row, last row, etc) enabled.

[*Example*: Consider the following example of whole table being used within DrawingML:

```
<wholeTbl>
  <tcTxStyle>
    …
  </tcTxStyle>
  <tcStyle>
    …
  </tcStyle>
</wholeTbl>
```

In this example, we see definitions for the text and the cells within the table. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.5 3D

The 3D portion of the DrawingML framework allows for the describing of a 3D scene to be placed within a document. This 3D scene can be described using text and shape objects along with various lighting, material and camera settings.

20.1.5.1 anchor (Anchor Point)

This element specifies a point in 3D space. This point is the point in space that anchors the backdrop plane. Please see the example in the backdrop (§20.1.5.2) definition for an in depth explanation of this element.

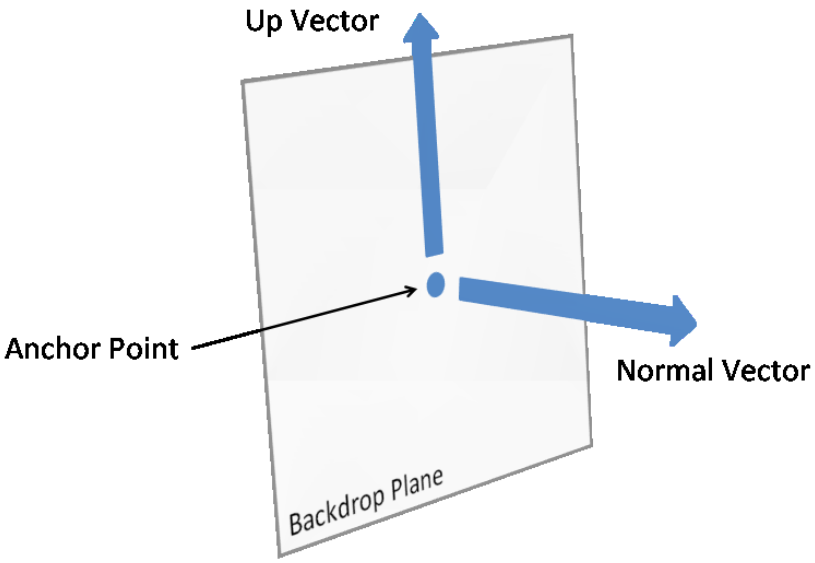
Attributes	Description
x (X-Coordinate in 3D)	X-Coordinate in 3D space. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
y (Y-Coordinate in 3D)	Y-Coordinate in 3D space. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
z (Z-Coordinate in 3D)	Z-Coordinate in 3D space. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Point3D) is located in §A.4.1. *end note*]

20.1.5.2 backdrop (Backdrop Plane)

This element defines a plane in which effects, such as glow and shadow, are applied in relation to the shape they are being applied to. The points and vectors contained within the backdrop define a plane in 3D space.

[*Example*: Consider the following image as an explanation of the backdrop plane definition:



In this image we see a plane being defined by an anchor point, the vector normal to the face of the plane and a vector pointing up in relation to the plane. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Backdrop) is located in §A.4.1. *end note*]

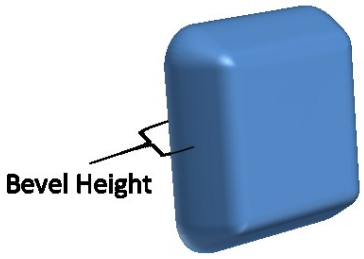
20.1.5.3 bevelB (Bottom Bevel)

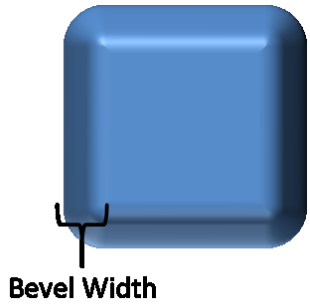
This element holds the properties associated with defining a bevel on the bottom or back face of a shape.

[*Example*: Consider the following example of an sp3d containing a bottom bevel.

```
<a:sp3d>
  <a:bevelB w="139700" h="127000" prst="coolSlant"/>
</a:sp3d>
```

In this example, we see a bottom bevel being defined with a preset bevel type along with a custom width and height. *end example*]

Attributes	Description
h (Height)	Specifies the height of the bevel, or how far above the shape it is applied. [*Example*: Consider the following example bevel  In this example, we see the height of an example bevel on a shape. *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
prst (Preset Bevel)	Specifies the preset bevel type which defines the look of the bevel. The possible values for this attribute are defined by the ST_BevelPresetType simple type (§20.1.10.9).
w (Width)	Specifies the width of the bevel, or how far into the shape it is applied. [*Example*: Consider the following example bevel

Attributes	Description
	 In this example, we see the width of an example bevel on a shape. *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Bevel) is located in §A.4.1. *end note*]

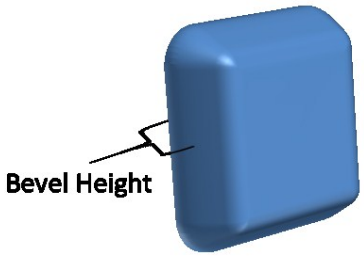
20.1.5.4 bevelT (Top Bevel)

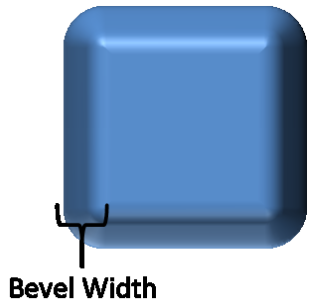
This element holds the properties associated with defining a bevel on the top or front face of a shape.

[*Example*: Consider the following example of an sp3d containing a top bevel.

```
<a:sp3d>
  <a:bevelT w="139700" h="127000" prst="coolSlant"/>
</a:sp3d>
```

In this example, we see a top bevel being defined with a preset bevel type along with a custom width and height. *end example*]

Attributes	Description
h (Height)	Specifies the height of the bevel, or how far above the shape it is applied. [*Example*: Consider the following example bevel  In this example, we see the height of an example bevel on a shape. *end example*]

Attributes	Description
	The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
prst (Preset Bevel)	Specifies the preset bevel type which defines the look of the bevel. The possible values for this attribute are defined by the ST_BevelPresetType simple type (§20.1.10.9).
w (Width)	Specifies the width of the bevel, or how far into the shape it is applied. [*Example*: Consider the following example bevel  In this example, we see the width of an example bevel on a shape. *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Bevel) is located in §A.4.1. *end note*]

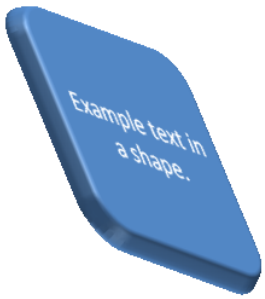
20.1.5.5 camera (Camera)

This element defines the placement and properties of the camera in the 3D scene. The camera position and properties modify the view of the scene.

[*Example*: Consider the following example of a camera in DrawingML:

```
<a:camera prst="orthographicFront">
  <a:rot lat="19902513" lon="17826689" rev="1362739"/>
</a:camera>
```

In this example, we see a preset camera being defined along with a rotation containing latitude, longitude, and revolution overrides provided that further rotate the camera around the scene. The effect of this camera can be seen on the following shape:



end example]

Attributes	Description
fov (Field of View)	Provides an override for the default field of view for the camera. Different perspectives can be obtained by modifying this attribute. [*Example*: Consider the following example of a fov in DrawingML: `<a:camera prst="perspectiveContrastingRightFacing" fov="6900000"> <a:rot lat="1200000" lon="18000000" rev="1200000"/> </a:camera>` In this example, we see a fov being defined which modifies the default fov for the preset camera. *end example*] The possible values for this attribute are defined by the ST_FOVAngle simple type (§20.1.10.26).
prst (Preset Camera Type)	Defines the preset camera that is being used by the camera element. The preset camera defines a starting point for common preset rotations in space. [*Example*: Consider the following example of a prst in DrawingML: `<a:camera prst="perspectiveContrastingRightFacing" fov="6900000"> <a:rot lat="1200000" lon="18000000" rev="1200000"/> </a:camera>` In this example, we see a prst being defined as `perspectiveContrastingRightFacing`. *end example*] The possible values for this attribute are defined by the ST_PresetCameraType simple type (§20.1.10.46).
zoom (Zoom)	Defines the zoom factor of a given camera element. The zoom modifies the scene as a whole and zooms in or out accordingly. [*Example*: Consider the following example of a zoom in DrawingML:

Attributes	Description
	<a:camera prst="perspectiveContrastingRightFacing" fov="6900000" zoom="200%"> <a:rot lat="1200000" lon="18000000" rev="1200000"/> /a:camera> In this example, we see a zoom being used which zooms the scene by 200%. *end example*] The possible values for this attribute are defined by the ST_PositivePercentage simple type (§20.1.10.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Camera) is located in §A.4.1. *end note*]

20.1.5.6 contourClr (Contour Color)

This element defines the color for the contour on a shape. The contour of a shape is a solid filled line which surrounds the outer edges of the shape.

[*Example*: Consider the following example of a contour defined on a shape which includes a contourClr. Lighting characteristics applied to the shape are ignored when it comes to the contour on the shape.

```
<a:sp3d contourW="101600" prstMaterial="plastic">
  <a:bevelT w="254000" h="254000"/>
  <a:bevelB w="254000" h="254000"/>
  <a:contourClr>
    <a:schemeClr val="bg1"/>
  </a:contourClr>
</a:sp3d>
```

In this example, we see a contour defined on a shape with a top and bottom bevel defined. In the image below, the contour is the white ring around the shape.

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.5.7 extrusionClr (Extrusion Color)

This element defines the color of the extrusion applied to a shape. The extrusion on a shape is an artificial height applied to the geometry.

[*Example*: Consider the following example of an extrusion which takes advantage of the extrusionClr. Lighting characteristics that are applied to the shape are also applied to the extrusion on the shape.

```
<a:sp3d extrusionH="139700" prstMaterial="plastic">
  <a:bevelT w="254000" h="254000"/>
  <a:bevelB w="254000" h="254000"/>
  <a:extrusionClr>
  <a:srgbClr val="FF0000"/>
  </a:extrusionClr>
</a:sp3d>
```

In this example, we see the extrusion color defined as red which can also be shown applied to the shape in the following image:

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.5.8 flatTx (No text in 3D scene)

Keep text out of 3D scene entirely.

Attributes	Description
z (Z Coordinate)	Specifies the Z coordinate to be used in positioning the flat text within the 3D scene. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FlatText) is located in §A.4.1. *end note*]

20.1.5.9 lightRig (Light Rig)

This element defines the light rig associated with the table. The light rig comes into play when there is a 3D bevel applied to a cell. When 3D is used, the light rig defines the lighting properties associated with the scene.

Attributes	Description
dir (Direction)	Defines the direction from which the light rig is oriented in relation to the scene. [*Example*: Consider the following example of dir being used in a light rig: `<a:lightRig rig="threePt" dir="t"/>` In this example, we define the direction to be `top`. *end example*]

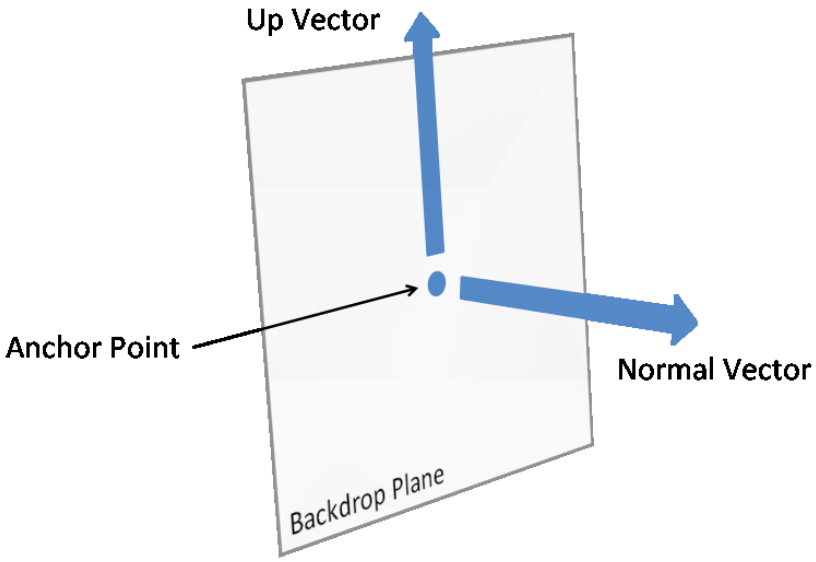
Attributes	Description
	The possible values for this attribute are defined by the ST_LightRigDirection simple type (§20.1.10.29).
rig (Rig Preset)	Defines the preset type of light rig which is to be applied to the scene. [*Example*: Consider the following example of rig being used in a light rig: `<a:lightRig rig="threePt" dir="t"/>` In this example, we define the rig to be a `threePt` rig. *end example*] The possible values for this attribute are defined by the ST_LightRigType simple type (§20.1.10.30).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LightRig) is located in §A.4.1. *end note*]

20.1.5.10 norm (Normal)

This element defines a normal vector. To be more precise, this attribute defines a vector normal to the face of the backdrop plane.

[*Example*: Consider the following image as an example of what a normal vector is in relation to the backdrop plane:



end example]

Attributes	Description
dx (Distance along X-axis in 3D)	Distance along X-axis in 3D The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
dy (Distance along Y-axis in 3D)	Distance along Y-axis in 3D The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
dz (Distance along Z-axis in 3D)	Distance along Z-axis in 3D The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Vector3D) is located in §A.4.1. *end note*]

20.1.5.11 rot (Rotation)

This element defines a rotation in 3D space. A rotation in DrawingML is defined through the use of a latitude coordinate, a longitude coordinate, and a revolution about the axis as the latitude and longitude coordinates.

[*Example*: Consider the following example of a rotation defined by the rot elements being used in a lightRig in DrawingML:

```
<a:lightRig rig="twoPt" dir="t">
  <a:rot lat="0" lon="0" rev="6000000"/>
</a:lightRig>
```

In this example, we have only a revolution applied to the light rig which rotates it around it's center axis. *end example*]

Attributes	Description
lat (Latitude)	Defines the latitude value of the rotation. [*Example*: Consider the following example of a rot in DrawingML: `<a:rot lat="0" lon="0" rev="6000000"/>` In this example, we set the lat to be equal to `0`. *end example*] The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.43).
lon (Longitude)	Defines the longitude value of the rotation.

Attributes	Description
	[*Example*: Consider the following example of a rot in DrawingML: `<a:rot lat="0" lon="0" rev="6000000"/>` In this example, we set the lon to be equal to `0`. *end example*] The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.43).
rev (Revolution)	This attributes defines the revolution around the central axis in the rotation. [*Example*: Consider the following example of a rot in DrawingML: `<a:rot lat="0" lon="0" rev="6000000"/>` In this example, we set the rev to be equal to `6000000`. *end example*] The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.43).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SphereCoords) is located in §A.4.1. *end note*]

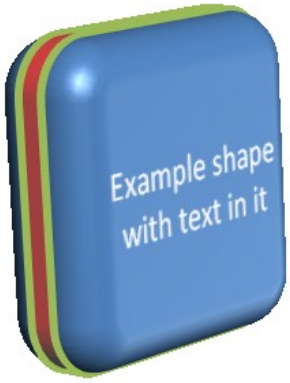
20.1.5.12 sp3d (Apply 3D shape properties)

This element defines the 3D properties associated with a particular shape in DrawingML. The 3D properties which can be applied to a shape are top and bottom bevels, a contour and an extrusion.

[*Example*: Consider the following example of an sp3d in DrawingML:

```
<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 an sp3d defined which contains information defining both a top and bottom bevel, along with an extrusion and contour on the shape. The following image illustrates a shape with the applied sp3d:



end example]

Attributes	Description
contourW (Contour Width)	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 (§20.1.10.41).
extrusionH (Extrusion Height)	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"/>`

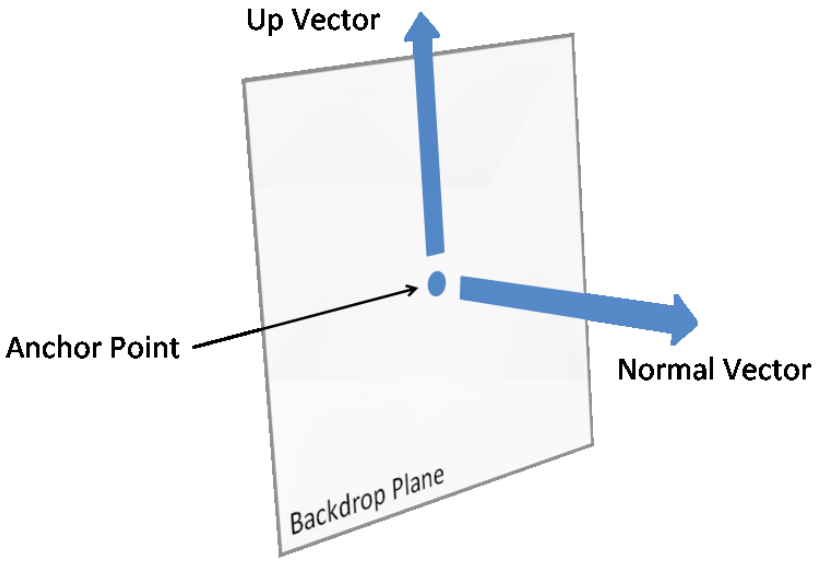
Attributes	Description
	</a:extrusionClr> <a:contourClr> <a:schemeClr val="accent3"/> </a:contourClr> </a:sp3d> In this example, we see a extrusionH defined as 165100. *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
prstMaterial (Preset Material Type)	Defines the preset material which 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 (§20.1.10.49).
z (Shape Depth)	Defines the z coordinate for the 3D shape. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Shape3D) is located in §A.4.1. *end note*]

20.1.5.13 up (Up Vector)

This element defines a vector representing up. To be more precise, this attribute defines a vector representing up in relation to the face of the backdrop plane.

[*Example*: Consider the following image as an example of what an up vector is in relation to the backdrop plane:



end example]

Attributes	Description
dx (Distance along X-axis in 3D)	Distance along X-axis in 3D The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
dy (Distance along Y-axis in 3D)	Distance along Y-axis in 3D The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
dz (Distance along Z-axis in 3D)	Distance along Z-axis in 3D The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Vector3D) is located in §A.4.1. *end note*]

20.1.6 Shared Style Sheet

The shared style sheet aspects contained within DrawingML are responsible for containing formatting options and styles which can be used by applications to define a certain look or feel to documents. The shared style sheet can be used by any document category ([*Note*: For example, a presentation. *end note*]) to pull visual information from which formats the document in a certain way, or theme. The shared style sheet contains information that is not document-category specific.

20.1.6.1 clrMap (Color Map)

This element specifics the color mapping layer which allows a user to define colors for background and text. This allows for swapping out of light/dark colors for backgrounds and the text on top of the background in order to maintain readability of the text On a deeper level, this specifies exactly which colors the first 12 values refer to in the color scheme.

[*Example*: Consider the following example of a color map in use:

```
<clrMap bg1="lt1" tx1="dk1" bg2="lt2" tx2="dk2" accent1="accent1"
  accent2="accent2" accent3="accent3" accent4="accent4" accent5="accent5"
  accent6="accent6" hlink="hlink" folHlink="folHlink"/>
```

In this example, we see that bg1 is mapped to lt1, tx1 is mapped to dk1, and so on. *end example*]

Attributes	Description
accent1 (Accent 1)	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 (§20.1.10.14).
accent2 (Accent 2)	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 (§20.1.10.14).
accent3 (Accent 3)	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 (§20.1.10.14).
accent4 (Accent 4)	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 (§20.1.10.14).
accent5 (Accent 5)	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 (§20.1.10.14).
accent6 (Accent 6)	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 (§20.1.10.14).
bg1 (Background 1)	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 (§20.1.10.14).
bg2 (Background 2)	Specifies a color defined which is associated as the second background color.

Attributes	Description
	The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
folHlink (Followed Hyperlink)	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 (§20.1.10.14).
hlink (Hyperlink)	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 (§20.1.10.14).
tx1 (Text 1)	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 (§20.1.10.14).
tx2 (Text 2)	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 (§20.1.10.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorMapping) is located in §A.4.1. *end note*]

20.1.6.2 clrScheme (Color Scheme)

This element defines a set of colors which are referred to as a color scheme. The color scheme is responsible for defining a list of twelve colors. The twelve colors consist of six accent colors, two dark colors, two light colors and a color for each of a hyperlink and followed hyperlink.

The Color Scheme Color elements appear in a sequence. The following listing shows the index value and corresponding Color Name.

Sequence Index	Element (Color) Name
0	dk1 (Dark 1)
1	lt1 (Light 1)
2	dk2 (Dark 2)
3	lt2 (Light 2)
4	accent1 (Accent 1)
5	accent2 (Accent 2)
6	accent3 (Accent 3)
7	accent4 (Accent 4)

Sequence Index	Element (Color) Name
8	accent5 (Accent 5)
9	accent6 (Accent 6)
10	hlink (Hyperlink)
11	folHlink (Followed Hyperlink)

[*Example*: Consider the following example of a color scheme defined in DrawingML:

```
<clrScheme name="sample">
  <dk1>
    <sysClr val="windowText"/>
  </dk1>
  <lt1>
    <sysClr val="window"/>
  </lt1>
  <dk2>
    <srgbClr val="04617B"/>
  </dk2>
  <lt2>
    <srgbClr val="DBF5F9"/>
  </lt2>
  <accent1>
    <srgbClr val="0F6FC6"/>
  </accent1>
  <accent2>
    <srgbClr val="009DD9"/>
  </accent2>
  <accent3>
    <srgbClr val="0BD0D9"/>
  </accent3>
  <accent4>
    <srgbClr val="10CF9B"/>
  </accent4>
  <accent5>
    <srgbClr val="7CCA62"/>
  </accent5>
  <accent6>
    <srgbClr val="A5C249"/>
  </accent6>
  <hlink>
    <srgbClr val="FF9800"/>
  </hlink>
```



```
    <folHlink>
      <srgbClr val="F45511"/>
    </folHlink>
  </clrScheme>
```

In this example, are defined the 12 theme colors in the sample color scheme. *end example*]

Attributes	Description
name (Name)	The common name for this color scheme. This name can show up in the user interface in a list of color schemes. 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_ColorScheme) is located in §A.4.1. *end note*]

20.1.6.3 custClrLst (Custom Color List)

This element allows for a custom color palette to be created and which shows up alongside other color schemes. This can be very useful, for example, when someone would like to maintain a corporate color palette.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomColorList) is located in §A.4.1. *end note*]

20.1.6.4 extraClrScheme (Extra Color Scheme)

This element defines an auxiliary color scheme, which includes both a color scheme and color mapping. This is mainly used for backward compatibility concerns and roundtrips information required by earlier versions.

[*Example*: Consider the following example of an extra color scheme in use in DrawingML:

```
<extraClrScheme>
  <clrScheme name="extraColorSchemeSample">
    <dk1>
      <sysClr val="windowText"/>
    </dk1>
    <lt1>
      <sysClr val="window"/>
    </lt1>
    <dk2>
      <srgbClr val="04617B"/>
    </dk2>
    <lt2>
      <srgbClr val="DBF5F9"/>
    </lt2>
```

```
      <accent1>
        <srgbClr val="0F6FC6"/>
      </accent1>
      <accent2>
        <srgbClr val="009DD9"/>
      </accent2>
      <accent3>
        <srgbClr val="0BD0D9"/>
      </accent3>
      <accent4>
        <srgbClr val="10CF9B"/>
      </accent4>
      <accent5>
        <srgbClr val="7CCA62"/>
      </accent5>
      <accent6>
        <srgbClr val="A5C249"/>
      </accent6>
      <hlink>
        <srgbClr val="FF9800"/>
      </hlink>
      <folHlink>
        <srgbClr val="F45511"/>
      </folHlink>
    </clrScheme>
    <clrMap bg1="lt1" tx1="dk1" bg2="lt2" tx2="dk2" accent1="accent1"
    accent2="accent2" accent3="accent3" accent4="accent4" accent5="accent5"
    accent6="accent6" hlink="hlink" folHlink="folHlink"/>
  </extraClrScheme>
```

In this example, the extra color scheme contains a color scheme and a color map for that color scheme. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorSchemeAndMapping) is located in §A.4.1. *end note*]

20.1.6.5 extraClrSchemeLst (Extra Color Scheme List)

This element is a container for the list of extra color schemes present in a document.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorSchemeList) is located in §A.4.1. *end note*]

20.1.6.6 masterClrMapping (Master Color Mapping)

This element is a part of a choice for which color mapping is used within the document. There is also defined an overrideClrMapping (§20.1.6.8) element which, when specified, the override is used rather than the color mapping defined in the master. If this element is specified, then we specifically use the color mapping defined in the master.

[*Note*: The W3C XML Schema definition of this element's content model (CT_EmptyElement) is located in §A.4.1. *end note*]

20.1.6.7 objectDefaults (Object Defaults)

This element allows for the definition of default shape, line, and textbox formatting properties. An application can use this information to format a shape (or text) initially on insertion into a document.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ObjectStyleDefaults) is located in §A.4.1. *end note*]

20.1.6.8 overrideClrMapping (Override Color Mapping)

This element provides an override for the color mapping in a document. When defined, this color mapping is used in place of the already defined color mapping, or master color mapping. This color mapping is defined in the same manner as the other mappings within this document.

[*Example*: Consider the following example of an override color mapping in DrawingML:

```
<overrideClrMapping bg1="lt1" tx1="dk1" bg2="lt2" tx2="dk2" accent1="accent1"
  accent2="accent2" accent3="accent3" accent4="accent4" accent5="accent5"
  accent6="accent6" hlink="hlink" folHlink="folHlink"/>
```

end example]

Attributes	Description
accent1 (Accent 1)	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 (§20.1.10.14).
accent2 (Accent 2)	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 (§20.1.10.14).
accent3 (Accent 3)	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 (§20.1.10.14).
accent4 (Accent 4)	Specifies a color defined which is associated as the accent 4 color.

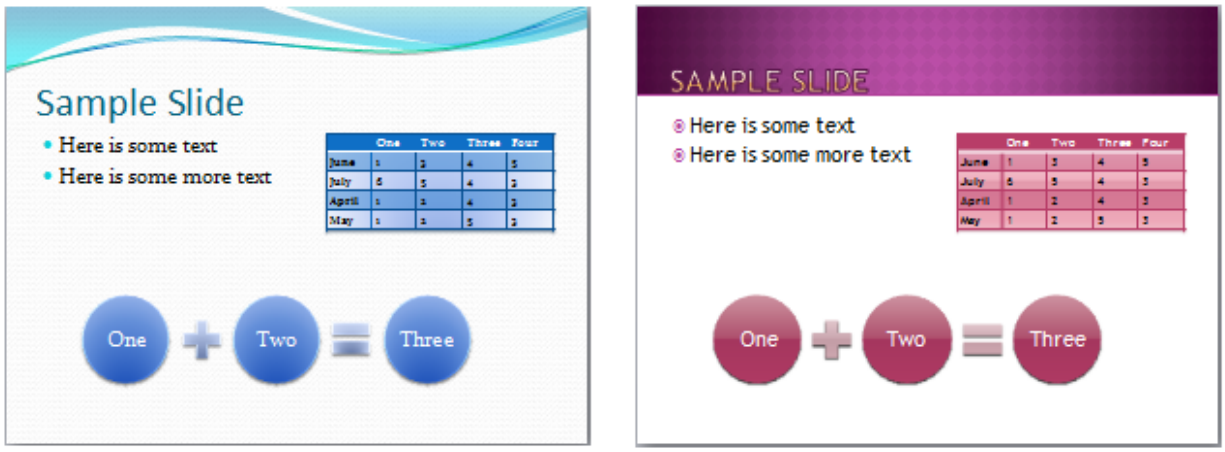
Attributes	Description
	The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent5 (Accent 5)	Specifies a color defined which is associated as the accent 5 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent6 (Accent 6)	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 (§20.1.10.14).
bg1 (Background 1)	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 (§20.1.10.14).
bg2 (Background 2)	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 (§20.1.10.14).
folHlink (Followed Hyperlink)	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 (§20.1.10.14).
hlink (Hyperlink)	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 (§20.1.10.14).
tx1 (Text 1)	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 (§20.1.10.14).
tx2 (Text 2)	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 (§20.1.10.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorMapping) is located in §A.4.1. *end note*]

20.1.6.9 theme (Theme)

This element defines the root level complex type associated with a shared style sheet (or theme). This element holds all the different formatting options available to a document through a theme and defines the overall look and feel of the document when themed objects are used within the document.

[*Example*: Consider the following image as an example of different themes in use applied to a presentation:



In this example, we see how a theme can affect font, colors, backgrounds, fills, and effects for different objects in a presentation. *end example*]

Attributes	Description
name (Name)	Specifies the name given to the theme. 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_OfficeStyleSheet) is located in §A.4.1. *end note*]

20.1.6.10 themeElements (Theme Elements)

This element defines the theme formatting options for the theme and is the workhorse of the theme. This is where the bulk of the shared theme information is contained and used by a document. This element contains the color scheme, font scheme, and format scheme elements which define the different formatting aspects of what a theme defines.

[*Example*: Consider the following example of a theme elements defined in DrawingML:

```
<themeElements>
  <clrScheme name="sample">
…
  </clrScheme>
  <fontScheme name="sample">
…
  </fontScheme>
  <fmtScheme name="sample">
    <fillStyleLst>
…
    </fillStyleLst>
    <lnStyleLst>
…
    </lnStyleLst>
    <effectStyleLst>
…
    </effectStyleLst>
    <bgFillStyleLst>
…
    </bgFillStyleLst>
  </fmtScheme>
</themeElements>
```

In this example, we see the basic structure of how a theme elements is defined and have left out the true guts of each individual piece to save room. Each part (color scheme, font scheme, format scheme) is defined elsewhere within DrawingML. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_BaseStyles) is located in §A.4.1. *end note*]

20.1.6.11 themeManager (Theme Manager)

The starting part for a theme file.

[*Note*: The W3C XML Schema definition of this element's content model (CT_EmptyElement) is located in §A.4.1. *end note*]

20.1.6.12 themeOverride (Theme Override)

This element allows for an override which changes just the colors, fonts, or effects of a single object, like a table for example. Currently it is used only to control overrides on the non-top-level masters within a presentation.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BaseStylesOverride) is located in §A.4.1. *end note*]

20.1.7 Coordinate Systems and Transformations

The following elements are used to reflect dimensions, scaling, location, rotation, and flip information on groups and individual shapes respectively.

20.1.7.1 chExt (Child Extents)

This element specifies the size dimensions of the child extents rectangle and is used for calculations of grouping, scaling, and rotation behavior of shapes placed within a group.

Attributes	Description
cx (Extent Length)	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 (§20.1.10.41).
cy (Extent Width)	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 (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositiveSize2D) is located in §A.4.1. *end note*]

20.1.7.2 chOff (Child Offset)

This element specifies the location of the child extents rectangle and is used for calculations of grouping, scaling, and rotation behavior of shapes placed within a group.

Attributes	Description
x (X-Axis	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified

Attributes	Description
Coordinate)	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 (§20.1.10.16).
y (Y-Axis Coordinate)	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 y attribute defines a y-coordinate of `100`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Point2D) is located in §A.4.1. *end note*]

20.1.7.3 ext (Extents)

This element specifies the size of the bounding box enclosing the referenced object.

Attributes	Description
cx (Extent Length)	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 (§20.1.10.41).

Attributes	Description
cy (Extent Width)	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 (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositiveSize2D) is located in §A.4.1. *end note*]

20.1.7.4 off (Offset)

This element specifies the location of the bounding box of an object. Effects on an object are not included in this bounding box.

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. [*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 (§20.1.10.16).
y (Y-Axis Coordinate)	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 y attribute defines a y-coordinate of `100`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type

Attributes	Description
	(§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Point2D) is located in §A.4.1. *end note*]

20.1.7.5 xfrm (2D Transform for Grouped Objects)

This element is nearly identical to the representation of 2-D transforms for ordinary shapes (§20.1.7.6). The only addition is a member to represent the Child offset and the Child extents.

Attributes	Description
flipH (Horizontal Flip)	Horizontal flip. When true, this attribute defines that the group 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)	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)	Rotation. Specifies the clockwise rotation of a group in 1/64000 of a degree. The possible values for this attribute are defined by the ST_Angle simple type (§20.1.10.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupTransform2D) is located in §A.4.1. *end note*]

20.1.7.6 xfrm (2D Transform for Individual Objects)

This element represents 2-D transforms for ordinary shapes.

Attributes	Description
flipH (Horizontal Flip)	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)	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)	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 (§20.1.10.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Transform2D) is located in §A.4.1. *end note*]

20.1.8 Shape Fills, Effects, and Line Properties

This portion of the DrawingML framework describes effects defining the visual appearance of shapes and lines. Shapes can be filled in a variety of ways, with images, solid colors, gradients, or pattern fills. In addition, several visual effects can alter the appearance of a shape, and multiple effects can be combined together. Lines also can have special properties defining how they are rendered, included a dashed appearance or decorations at the line ends. This section documents the elements that define these properties and effects for shapes and lines.

20.1.8.1 alphaBiLevel (Alpha Bi-Level Effect)

This element represents an Alpha Bi-Level Effect.

Alpha (Opacity) values less than the threshold are changed to 0 (fully transparent) and alpha values greater than or equal to the threshold are changed to 100% (fully opaque).

Attributes	Description
thresh (Threshold)	Specifies the threshold value for the alpha bi-level effect. The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.44).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AlphaBiLevelEffect) is located in §A.4.1. *end note*]

20.1.8.2 alphaCeiling (Alpha Ceiling Effect)

This element represents an alpha ceiling effect.

Alpha (opacity) values greater than zero are changed to 100%. In other words, anything partially opaque becomes fully opaque.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AlphaCeilingEffect) is located in §A.4.1. *end note*]

20.1.8.3 alphaFloor (Alpha Floor Effect)

This element represents an alpha floor effect.

Alpha (opacity) values less than 100% are changed to zero. In other words, anything partially transparent becomes fully transparent.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AlphaFloorEffect) is located in §A.4.1. *end note*]

20.1.8.4 alphaInv (Alpha Inverse Effect)

This element represents an alpha inverse effect.

Alpha (opacity) values are inverted by subtracting from 100%.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AlphaInverseEffect) is located in §A.4.1. *end note*]

20.1.8.5 alphaMod (Alpha Modulate Effect)

This element represents an alpha modulate effect.

Effect alpha (opacity) values are multiplied by a fixed percentage. The effect container specifies an effect containing alpha values to modulate.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AlphaModulateEffect) is located in §A.4.1. *end note*]

20.1.8.6 alphaModFix (Alpha Modulate Fixed Effect)

This element represents an alpha modulate fixed effect.

Effect alpha (opacity) values are multiplied by a fixed percentage.

Attributes	Description
amt (Amount)	Specifies the percentage amount to scale the alpha. The possible values for this attribute are defined by the ST_PositivePercentage simple type (§20.1.10.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AlphaModulateFixedEffect) is located in §A.4.1. *end note*]

20.1.8.7 alphaOutset (Alpha Inset/Outset Effect)

This element specifies an alpha outset/inset effect.

This is equivalent to an alpha ceiling, followed by alpha blur, followed by either an alpha ceiling (positive radius) or alpha floor (negative radius).

Attributes	Description
rad (Radius)	Specifies the radius of outset/inset. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AlphaOutsetEffect) is located in §A.4.1. *end note*]

20.1.8.8 alphaRepl (Alpha Replace Effect)

This element specifies an alpha replace effect.

Effect alpha (opacity) values are replaced by a fixed alpha.

Attributes	Description
a (Alpha)	Specifies the new opacity value. The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.44).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AlphaReplaceEffect) is located in §A.4.1. *end note*]

20.1.8.9 bevel (Line Join Bevel)

This element specifies a Bevel Line Join.

A bevel joint specifies that an angle joint is used to connect lines.

[*Example*:

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineJoinBevel) is located in §A.4.1. *end note*]

20.1.8.10 bgClr (Background color)

This element specifies the background color of a Pattern fill.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.8.11 biLevel (Bi-Level (Black/White) Effect)

This element specifies a bi-level (black/white) effect. Input colors whose luminance is less than the specified threshold value are changed to black. Input colors whose luminance are greater than or equal the specified value are set to white. The alpha effect values are unaffected by this effect.

Attributes	Description
thresh (Threshold)	Specifies the luminance threshold for the Bi-Level effect. Values greater than or equal to the threshold are set to white. Values lesser than the threshold are set to black.

Attributes	Description
	The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.44).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BiLevelEffect) is located in §A.4.1. *end note*]

20.1.8.12 blend (Blend Effect)

This element specifies a blend of several effects. The container specifies the raw effects to blend while the blend mode specifies how the effects are to be blended.

Attributes	Description
blend (Blend Mode)	Specifies how to blend the two effects. The possible values for this attribute are defined by the ST_BlendMode simple type (§20.1.10.11).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BlendEffect) is located in §A.4.1. *end note*]

20.1.8.13 blip (Blip)

This element specifies the existence of an image (binary large image or picture) and contains a reference to the image data.

Attributes	Description
cstate (Compression State)	Specifies the compression state with which the picture is stored. This allows the application to specify the amount of compression that has been applied to a picture. The possible values for this attribute are defined by the ST_BlipCompression simple type (§20.1.10.12).
embed (Embedded Picture Reference) Namespace: http://purl.oclc.org/ooxml/officeDocument/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 (§22.8.2.1).
link (Linked Picture Reference)	Specifies the identification information for a linked picture. This attribute is used to specify an image that does not reside within this file.

Attributes	Description
Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Blip) is located in §A.4.1. *end note*]

20.1.8.14 blipFill (Picture Fill)

This element specifies the type of picture fill that the picture object has. Because a picture has a picture fill already by default, it is possible to have two fills specified for a picture object. An example of this is shown below.

[*Example*: Consider the picture below that has a blip fill applied to it. The image used to fill this picture object has transparent pixels instead of white pixels.

```
<p:pic>
  …
  <p:blipFill>
    <a:blip r:embed="rId2"/>
    <a:stretch>
      <a:fillRect/>
    </a:stretch>
  </p:blipFill>
  …
</p:pic>
```

The above picture object is shown as an example of this fill type. *end example*]

[*Example*: Consider now the same picture object but with an additional gradient fill applied within the shape properties portion of the picture.

```
<p:pic>
  …
  <p:blipFill>
    <a:blip r:embed="rId2"/>
    <a:stretch>
      <a:fillRect/>
    </a:stretch>
  </p:blipFill>
  <p:spPr>
    <a:gradFill>
      <a:gsLst>
        <a:gs pos="0">
          <a:schemeClr val="tx2">
            <a:shade val="50000"/>
          </a:schemeClr>
        </a:gs>
        <a:gs pos="39999">
          <a:schemeClr val="tx2">
            <a:tint val="20000"/>
          </a:schemeClr>
        </a:gs>
        <a:gs pos="70000">
          <a:srgbClr val="C4D6EB"/>
        </a:gs>
        <a:gs pos="100000">
          <a:schemeClr val="bg1"/>
        </a:gs>
      </a:gsLst>
    </a:gradFill>
  </p:spPr>
  …
</p:pic>
```

The above picture object is shown as an example of this double fill type. *end example*]

Attributes	Description
dpi (DPI Setting)	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)	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.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BlipFillProperties) is located in §A.4.1. *end note*]

20.1.8.15 blur (Blur Effect)

This element specifies a blur effect that is applied to the entire shape, including its fill. All color channels, including alpha, are affected.

Attributes	Description
grow (Grow Bounds)	Specifies whether the bounds of the object should be grown as a result of the blurring. True indicates the bounds are grown while false indicates that they are not. [*Example*:

Attributes	Description
	With grow set to false, the blur effect does not extend beyond the original bounds of the object:  With grow set to true, the blur effect can extend beyond the original bounds of the object:  *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rad (Radius)	Specifies the radius of blur. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BlurEffect) is located in §A.4.1. *end note*]

20.1.8.16 clrChange (Color Change Effect)

This element specifies a Color Change Effect. Instances of clrFrom are replaced with instances of clrTo.

Attributes	Description
useA (Consider Alpha Values)	Specifies whether alpha values are considered for the effect. Effect alpha values are considered if useA is true, else they are ignored. 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_ColorChangeEffect) is located in §A.4.1. *end note*]

20.1.8.17 clrFrom (Change Color From)

This element specifies a color getting removed in a color change effect. It is the "from" or source input color.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.8.18 clrRepl (Solid Color Replacement)

This element specifies a solid color replacement value. All effect colors are changed to a fixed color. Alpha values are unaffected.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorReplaceEffect) is located in §A.4.1. *end note*]

20.1.8.19 clrTo (Change Color To)

This element specifies the color which replaces the clrFrom in a clrChange effect. This is the "target" or "to" color in the color change effect.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.8.20 cont (Effect Container)

This element specifies an Effect Container. It is a list of effects.

Attributes	Description
name (Name)	Specifies an optional name for this list of effects, so that it can be referred to later. Shall be unique across all effect trees and effect containers. The possible values for this attribute are defined by the W3C XML Schema token datatype.
type (Effect Container Type)	Specifies the kind of container, either sibling or tree. The possible values for this attribute are defined by the ST_EffectContainerType simple type (§20.1.10.22).

[*Note*: The W3C XML Schema definition of this element's content model (CT_EffectContainer) is located in §A.4.1. *end note*]

20.1.8.21 custDash (Custom Dash)

This element specifies a custom dashing scheme. It is a list of dash stop elements which represent building block atoms upon which the custom dashing scheme is built.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DashStopList) is located in §A.4.1. *end note*]

20.1.8.22 ds (Dash Stop)

This element specifies a dash stop primitive. Dashing schemes are built by specifying an ordered list of dash stop primitive. A dash stop primitive consists of a dash and a space.

Attributes	Description
d (Dash Length)	Specifies the length of the dash relative to the line width. The possible values for this attribute are defined by the ST_PositivePercentage simple type (§20.1.10.45).
sp (Space Length)	Specifies the length of the space relative to the line width. The possible values for this attribute are defined by the ST_PositivePercentage simple type (§20.1.10.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DashStop) is located in §A.4.1. *end note*]

20.1.8.23 duotone (Duotone Effect)

This element specifies a duotone effect.

For each pixel, combines clr1 and clr2 through a linear interpolation to determine the new color for that pixel.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DuotoneEffect) is located in §A.4.1. *end note*]

20.1.8.24 effect (Effect)

This element specifies a reference to an existing effect container.

Attributes	Description
ref (Reference)	Specifies the reference. Its value can be the name of an effect container, or one of four special references: fill - refers to the fill effect line - refers to the line effect fillLine - refers to the combined fill and line effects children - refers to the combined effects from logical child shapes or text The possible values for this attribute are defined by the W3C XML Schema token datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_EffectReference) is located in §A.4.1. *end note*]

20.1.8.25 effectDag (Effect Container)

This element specifies a list of effects. Effects are applied in the order specified by the container type (sibling or tree).

[*Note:* An effectDag element can contain multiple effect containers as child elements. Effect containers with different styles can be combined in an effectDag to define a directed acyclic graph (DAG) that specifies the order in which all effects are applied. *end note*]

Attributes	Description
name (Name)	Specifies an optional name for this list of effects, so that it can be referred to later. Shall be unique across all effect trees and effect containers. The possible values for this attribute are defined by the W3C XML Schema token datatype.
type (Effect Container Type)	Specifies the kind of container, either sibling or tree. The possible values for this attribute are defined by the ST_EffectContainerType simple type (§20.1.10.22).

[*Note*: The W3C XML Schema definition of this element's content model (CT_EffectContainer) is located in §A.4.1. *end note*]

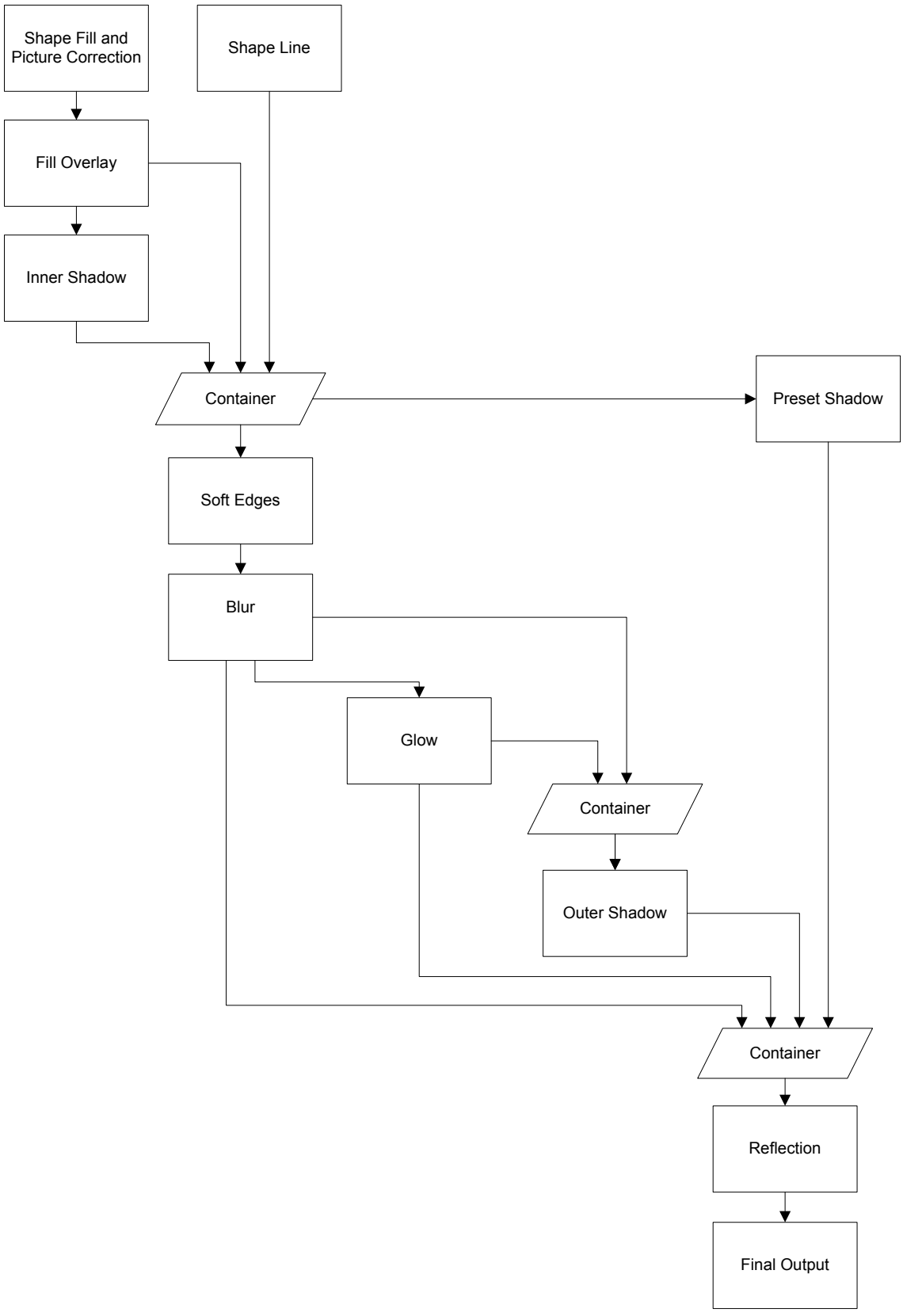
20.1.8.26 effectLst (Effect Container)

This element specifies a list of effects. Effects in an effectLst are applied in the default order by the rendering engine. The following diagrams illustrate the order in which effects are applied, both for shapes and for group shapes.

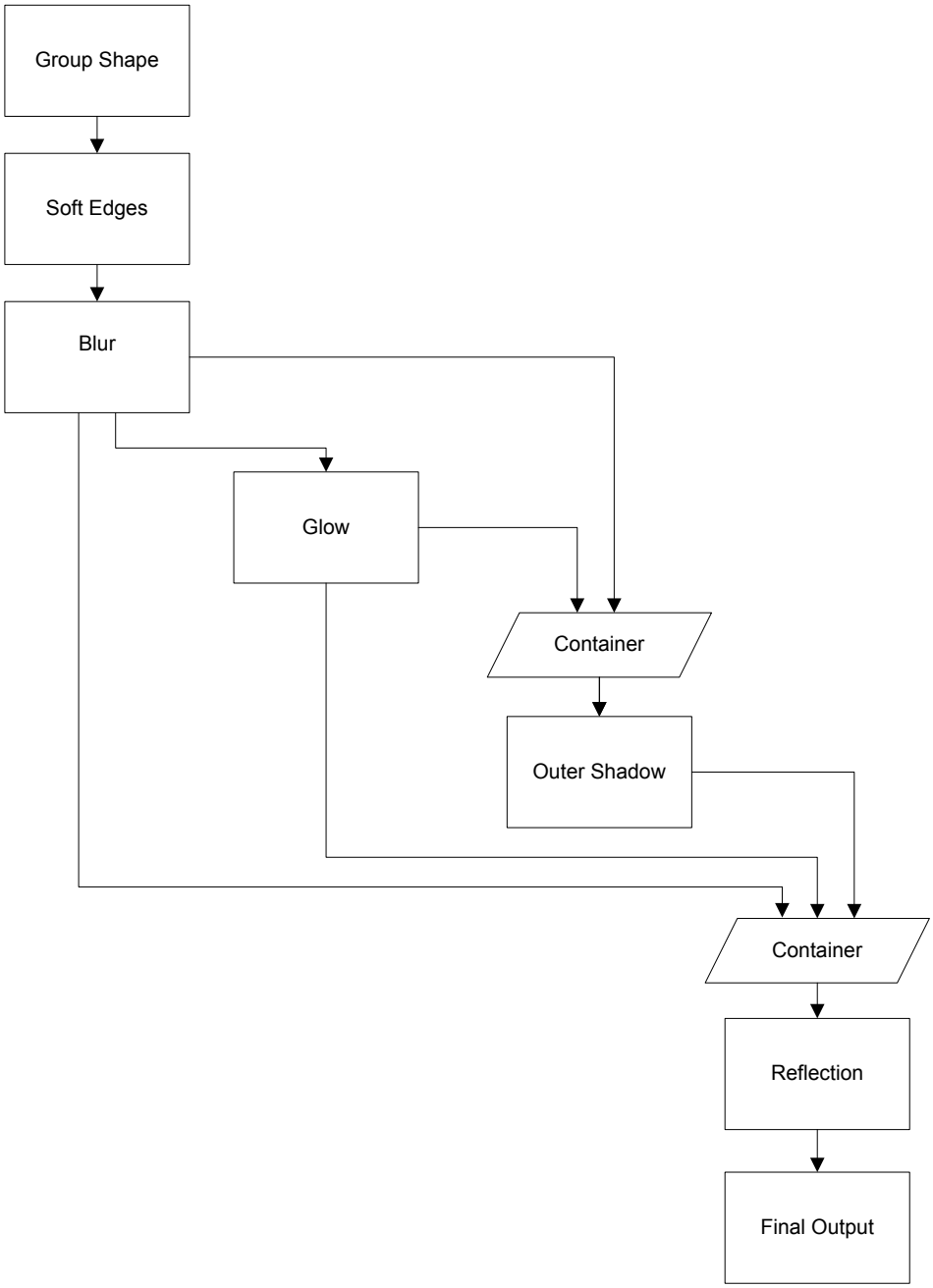
[*Note:* The output of many effects does not include the input shape. For effects that should be applied to the result of previous effects as well as the original shape, a container is used to group the inputs together. *end note*]

[*Example:* Outer Shadow is applied both to the original shape and the original shape's glow. The result of blur contains the original shape, while the result of glow contains only the added glow. Therefore, a container that groups the blur result with the glow result is used as the input to Outer Shadow. *end example*]

effectLst Processing for Shapes



effectLst Processing for Group Shapes



[*Note*: The W3C XML Schema definition of this element's content model (CT_EffectList) is located in §A.4.1. *end note*]

20.1.8.27 fgClr (Foreground color)

This element specifies the foreground color of a pattern fill.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.8.28 fill (Fill)

This element specifies a fill which is one of blipFill, gradFill, grpFill, noFill, pattFill or solidFill.

[*Note*: The W3C XML Schema definition of this element's content model (CT_FillEffect) is located in §A.4.1. *end note*]

20.1.8.29 fillOverlay (Fill Overlay Effect)

This element specifies a fill overlay effect. A fill overlay can be used to specify an additional fill for an object and blend the two fills together.

Attributes	Description
blend (Blend)	Specifies how to blend the fill with the base effect. The possible values for this attribute are defined by the ST_BlendMode simple type (§20.1.10.11).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FillOverlayEffect) is located in §A.4.1. *end note*]

20.1.8.30 fillRect (Fill Rectangle)

This element specifies a fill rectangle. When stretching of an image is specified, a source rectangle, srcRect, is scaled to fit the specified fill rectangle.

Each edge of the fill rectangle is defined by a percentage offset from the corresponding edge of the shape's bounding box. A positive percentage specifies an inset, while a negative percentage specifies an outset. [*Note*: For example, a left offset of 25% specifies that the left edge of the fill rectangle is located to the right of the bounding box's left edge by an amount equal to 25% of the bounding box's width. *end note*]

[*Example*:

```
<a:blipFill>
  <a:blip r:embed="rId2"/>
  <a:stretch>
      <a:fillRect b="10000" r="25000"/>
```

```
    </a:stretch>
  </a:blipFill>
```

The above image is stretched to fill the entire rectangle except for the bottom 10% and rightmost 25%.

end example]

Attributes	Description
b (Bottom Offset)	Specifies the bottom edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
l (Left Offset)	Specifies the left edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
r (Right Offset)	Specifies the right edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
t (Top Offset)	Specifies the top edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

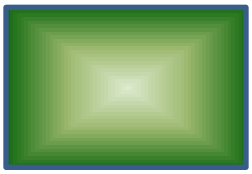
[*Note*: The W3C XML Schema definition of this element's content model (CT_RelativeRect) is located in §A.4.1. *end note*]

20.1.8.31 fillToRect (Fill To Rectangle)

This element defines the "focus" rectangle for the center shade, specified relative to the fill tile rectangle. The center shade fills the entire tile except the margins specified by each attribute.

Each edge of the center shade rectangle is defined by a percentage offset from the corresponding edge of the tile rectangle. A positive percentage specifies an inset, while a negative percentage specifies an outset. [*Note:* For example, a left offset of 25% specifies that the left edge of the center shade rectangle is located to the right of the tile rectangle's left edge by an amount equal to 25% of the tile rectangle's width. *end note*]

[*Example*:

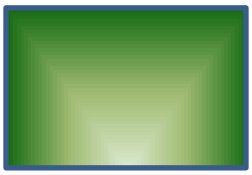



```
<a:path path="rect">
  <a:fillToRect l="50000" r="50000" t="50000" b="50000"/>
</a:path>
```

In the above shape, the rectangle defined by fillToRect is a single point in the center of the shape. This creates the effect of the center shade focusing at a point in the center of the region.

end example]

[*Example*:



```
<a:path path="rect">
  <a:fillToRect l="25000" t="25000" r="25000" b="0"/>
</a:path>
```

The center shade occupies the rectangle defined by excluding the topmost, leftmost, and rightmost 25% of the region. Therefore, the gradient fills the remaining leftmost 25%, topmost 25%, and rightmost 25% of the region.

end example]

Attributes	Description
b (Bottom Offset)	Specifies the bottom edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
l (Left Offset)	Specifies the left edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
r (Right Offset)	Specifies the right edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
t (Top Offset)	Specifies the top edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RelativeRect) is located in §A.4.1. *end note*]

20.1.8.32 glow (Glow Effect)

This element specifies a glow effect, in which a color blurred outline is added outside the edges of the object.

Attributes	Description
rad (Radius)	Specifies the radius of the glow. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GlowEffect) is located in §A.4.1. *end note*]

20.1.8.33 gradFill (Gradient Fill)

This element defines a gradient fill.

A gradient fill is a fill which is characterized by a smooth gradual transition from one color to the next. At its simplest, it is a fill which transitions between two colors; or more generally, it can be a transition of any number of colors.

The desired transition colors and locations are defined in the gradient stop list (gsLst) child element.

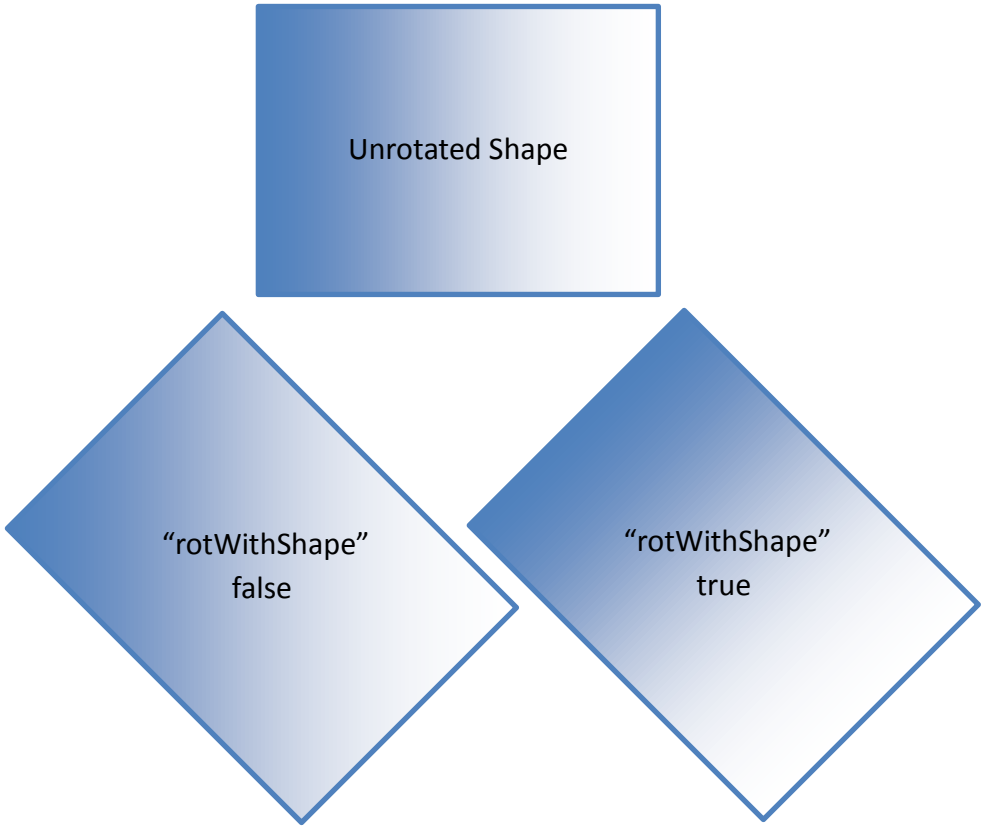
The other child element defines the properties of the gradient fill (there are two styles-- a linear shade style as well as a path shade style)

[*Example*:

The following is a sample gradient fill, varying from blue to white:

end example]

Attributes	Description
flip (Tile Flip)	Specifies the direction(s) in which to flip the gradient while tiling. Normally a gradient fill encompasses the entire bounding box of the shape which contains the fill. However, with the tileRect element, it is possible to define a "tile" rectangle which is smaller than the bounding box. In this situation, the gradient fill is encompassed within the tile rectangle, and the tile rectangle is tiled across the bounding box to fill the entire area.

Attributes	Description
	The possible values for this attribute are defined by the ST_TileFlipMode simple type (§20.1.10.85).
rotWithShape (Rotate With Shape)	Specifies if a fill rotates along with a shape when the shape is rotated. [*Example*: The following is a fill with the flip attribute set to "x". The black interior rectangle indicates the tile rectangle. Notice that the adjacent rectangle to the right in the tile has been flipped along the x-axis.  *end example*] 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_GradientFillProperties) is located in §A.4.1. *end note*]

20.1.8.34 grayscl (Gray Scale Effect)

This element specifies a gray scale effect. Converts all effect color values to a shade of gray, corresponding to their luminance. Effect alpha (opacity) values are unaffected.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GrayscaleEffect) is located in §A.4.1. *end note*]

20.1.8.35 grpFill (Group Fill)

This element specifies a group fill. When specified, this setting indicates that the parent element is part of a group and should inherit the fill properties of the group.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupFillProperties) is located in §A.4.1. *end note*]

20.1.8.36 gs (Gradient stops)

This element defines a gradient stop. A gradient stop consists of a position where the stop appears in the color band.

Attributes	Description
pos (Position)	Specifies where this gradient stop should appear in the color band. This position is specified in the range [0%, 100%], which corresponds to the beginning and the end of the color band respectively. The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.44).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GradientStop) is located in §A.4.1. *end note*]

20.1.8.37 gsLst (Gradient Stop List)

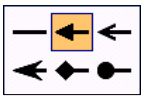
The list of gradient stops that specifies the gradient colors and their relative positions in the color band.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GradientStopList) is located in §A.4.1. *end note*]

20.1.8.38 headEnd (Line Head/End Style)

This element specifies decorations which can be added to the head of a line.

[*Example*:



end exmaple]

Attributes	Description
len (Length of Head/End)	Specifies the line end length in relation to the line width. The possible values for this attribute are defined by the ST_LineEndLength simple type (§20.1.10.32).
type (Line Head/End Type)	Specifies the line end decoration, such as a triangle or arrowhead. The possible values for this attribute are defined by the ST_LineEndType simple type (§20.1.10.33).
w (Width of Head/End)	Specifies the line end width in relation to the line width. The possible values for this attribute are defined by the ST_LineEndWidth simple type (§20.1.10.34).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineEndProperties) is located in §A.4.1. *end note*]

20.1.8.39 hsl (Hue Saturation Luminance Effect)

This element specifies a hue/saturation/luminance effect. The hue, saturation, and luminance can each be adjusted relative to its current value.

Attributes	Description
hue (Hue)	Specifies the number of degrees by which the hue is adjusted. The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.43).
lum (Luminance)	Specifies the percentage by which the luminance is adjusted. The possible values for this attribute are defined by the ST_FixedPercentage simple type (§20.1.10.24).
sat (Saturation)	Specifies the percentage by which the saturation is adjusted. The possible values for this attribute are defined by the ST_FixedPercentage simple type (§20.1.10.24).

[*Note*: The W3C XML Schema definition of this element's content model (CT_HSLEffect) is located in §A.4.1. *end note*]

20.1.8.40 innerShdw (Inner Shadow Effect)

This element specifies an inner shadow effect. A shadow is applied within the edges of the object according to the parameters given by the attributes.

Attributes	Description
blurRad (Blur Radius)	Specifies the blur radius. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
dir (Direction)	Specifies the direction to offset the shadow. The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.43).
dist (Distance)	Specifies how far to offset the shadow. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_InnerShadowEffect) is located in §A.4.1. *end note*]

20.1.8.41 lin (Linear Gradient Fill)

This element specifies a linear gradient.

Attributes	Description
ang (Angle)	Specifies the direction of color change for the gradient. To define this angle, let its value be x measured clockwise. Then (-sin x, cos x) is a vector parallel to the line of constant color in the gradient fill. The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.43).
scaled (Scaled)	Whether the gradient angle scales with the fill region. Mathematically, if this flag is true, then the gradient vector (cos x , sin x) is scaled by the width (w) and height (h) of the fill region, so that the vector becomes (w cos x, h sin x) (before normalization). Observe that now if the gradient angle is 45 degrees, the gradient vector is (w, h), which goes

Attributes	Description
	from top-left to bottom-right of the fill region. If this flag is false, the gradient angle is independent of the fill region and is not scaled using the manipulation described above. So a 45-degree gradient angle always give a gradient band whose line of constant color is parallel to the vector (1, -1). 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_LinearShadeProperties) is located in §A.4.1. *end note*]

20.1.8.42 lum (Luminance Effect)

This element specifies a luminance effect. Brightness linearly shifts all colors closer to white or black. Contrast scales all colors to be either closer or further apart.

Attributes	Description
bright (Brightness)	Specifies the percent to change the brightness. The possible values for this attribute are defined by the ST_FixedPercentage simple type (§20.1.10.24).
contrast (Contrast)	Specifies the percent to change the contrast. The possible values for this attribute are defined by the ST_FixedPercentage simple type (§20.1.10.24).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LuminanceEffect) is located in §A.4.1. *end note*]

20.1.8.43 miter (Miter Line Join)

This element specifies that a line join shall be mitered.

[*Example*: The following sample illustrated two lines which are joined using a mitered style

end example]

Attributes	Description
lim (Miter Join	Specifies the amount by which lines is extended to form a miter join - otherwise miter

Attributes	Description
Limit)	joins can extend infinitely far (for lines which are almost parallel). The possible values for this attribute are defined by the ST_PositivePercentage simple type (§20.1.10.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineJoinMiterProperties) is located in §A.4.1. *end note*]

20.1.8.44 noFill (No Fill)

This element specifies that no fill is applied to the parent element.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NoFillProperties) is located in §A.4.1. *end note*]

20.1.8.45 outerShdw (Outer Shadow Effect)

This element specifies an Outer Shadow Effect.

[*Example*: The following is an example of an outer shadow effect.

end example]

Attributes	Description
algn (Shadow Alignment)	Specifies shadow alignment; alignment happens first, effectively setting the origin for scale, skew, and offset. The possible values for this attribute are defined by the ST_RectAlignment simple type (§20.1.10.52).
blurRad (Blur Radius)	Specifies the blur radius of the shadow. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
dir (Shadow Direction)	Specifies the direction to offset the shadow. The possible values for this attribute are defined by the ST_PositiveFixedAngle simple

Attributes	Description
	type (§20.1.10.43).
dist (Shadow Offset Distance)	Specifies the how far to offset the shadow. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
kx (Horizontal Skew)	Specifies the horizontal skew angle. The possible values for this attribute are defined by the ST_FixedAngle simple type (§20.1.10.23).
ky (Vertical Skew)	Specifies the vertical skew angle. The possible values for this attribute are defined by the ST_FixedAngle simple type (§20.1.10.23).
rotWithShape (Rotate With Shape)	Specifies whether the shadow rotates with the shape if the shape is rotated. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sx (Horizontal Scaling Factor)	Specifies the horizontal scaling factor; negative scaling causes a flip. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
sy (Vertical Scaling Factor)	Specifies the vertical scaling factor; negative scaling causes a flip. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OuterShadowEffect) is located in §A.4.1. *end note*]

20.1.8.46 path (Path Gradient)

This element defines that a gradient fill follows a path vs. a linear line.

[*Example*:

The examples above illustrate gradient fills following a circular, rectangular or shape path.

end example]

Attributes	Description
path (Gradient Fill Path)	Specifies the shape of the path to follow. The possible values for this attribute are defined by the ST_PathShadeType simple type (§20.1.10.38).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PathShadeProperties) is located in §A.4.1. *end note*]

20.1.8.47 pattFill (Pattern Fill)

This element specifies a pattern fill. A repeated pattern is used to fill the object.

Attributes	Description
prst (Preset Pattern)	Specifies one of a set of preset patterns to fill the object. The possible values for this attribute are defined by the ST_PresetPatternVal simple type (§20.1.10.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PatternFillProperties) is located in §A.4.1. *end note*]

20.1.8.48 prstDash (Preset Dash)

This element specifies that a preset line dashing scheme should be used.

Attributes	Description
val (Value)	Specifies which preset dashing scheme is to be used. The possible values for this attribute are defined by the ST_PresetLineDashVal simple type (§20.1.10.48).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PresetLineDashProperties) is located in §A.4.1. *end note*]

20.1.8.49 prstShdw (Preset Shadow)

This element specifies that a preset shadow is to be used. Each preset shadow is equivalent to a specific outer shadow effect. For each preset shadow, the color element, direction attribute, and distance attribute represent the color, direction, and distance parameters of the corresponding outer shadow. Additionally, the rotateWithShape attribute of corresponding outer shadow is always false. Other non-default parameters of the outer shadow are dependent on the prst attribute.

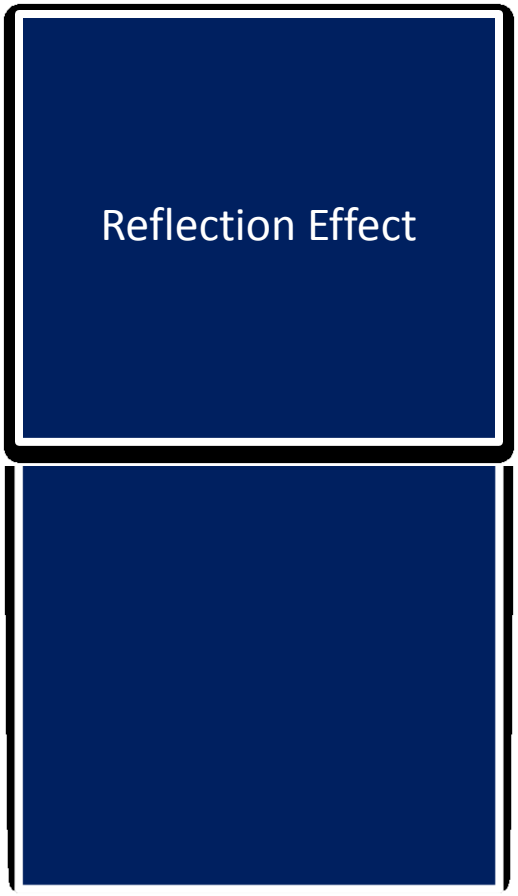
Attributes	Description
dir (Direction)	Specifies the direction to offset the shadow. The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.43).
dist (Distance)	Specifies how far to offset the shadow. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
prst (Preset Shadow)	Specifies which preset shadow to use. The possible values for this attribute are defined by the ST_PresetShadowVal simple type (§20.1.10.51).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PresetShadowEffect) is located in §A.4.1. *end note*]

20.1.8.50 reflection (Reflection Effect)

This element specifies a reflection effect.

[*Example*:



end example]

Attributes	Description
algn (Shadow Alignment)	Specifies shadow alignment. The possible values for this attribute are defined by the ST_RectAlignment simple type (§20.1.10.52).
blurRad (Blur Radius)	Specifies the blur radius. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
dir (Direction)	Specifies the direction of the alpha gradient ramp relative to the shape itself. The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.43).
dist (Distance)	Specifies how far to distance the shadow. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
endA (End Alpha)	Specifies the ending reflection opacity.

Attributes	Description
	The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.44).
endPos (End Position)	Specifies the end position (along the alpha gradient ramp) of the end alpha value. The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.44).
fadeDir (Fade Direction)	Specifies the direction to offset the reflection. The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.43).
kx (Horizontal Skew)	Specifies the horizontal skew angle. The possible values for this attribute are defined by the ST_FixedAngle simple type (§20.1.10.23).
ky (Vertical Skew)	Specifies the vertical skew angle. The possible values for this attribute are defined by the ST_FixedAngle simple type (§20.1.10.23).
rotWithShape (Rotate With Shape)	Specifies if the reflection rotates with the shape. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
stA (Start Opacity)	starting reflection opacity. The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.44).
stPos (Start Position)	Specifies the start position (along the alpha gradient ramp) of the start alpha value. The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.44).
sx (Horizontal Ratio)	Specifies the horizontal scaling factor. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
sy (Vertical Ratio)	Specifies the vertical scaling factor. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ReflectionEffect) is located in §A.4.1. *end note*]

20.1.8.51 relOff (Relative Offset Effect)

This element specifies a relative offset effect. Sets up a new origin by offsetting relative to the size of the previous effect.

Attributes	Description
tx (Offset X)	Specifies the X offset. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
ty (Offset Y)	Specifies the Y offset. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RelativeOffsetEffect) is located in §A.4.1. *end note*]

20.1.8.52 round (Round Line Join)

This element specifies that lines joined together have a round join.

[*Example*:

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineJoinRound) is located in §A.4.1. *end note*]

20.1.8.53 softEdge (Soft Edge Effect)

This element specifies a soft edge effect. The edges of the shape are blurred, while the fill is not affected.

Attributes	Description
rad (Radius)	Specifies the radius of blur to apply to the edges. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SoftEdgesEffect) is located in §A.4.1. *end note*]

20.1.8.54 solidFill (Solid Fill)

This element specifies a solid color fill. The shape is filled entirely with the specified color.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SolidColorFillProperties) is located in §A.4.1. *end note*]

20.1.8.55 srcRect (Source Rectangle)

This element specifies the portion of the blip used for the fill.

Each edge of the source rectangle is defined by a percentage offset from the corresponding edge of the bounding box. A positive percentage specifies an inset, while a negative percentage specifies an outset. [*Note:* For example, a left offset of 25% specifies that the left edge of the source rectangle is located to the right of the bounding box's left edge by an amount equal to 25% of the bounding box's width. *end note*]

Attributes	Description
b (Bottom Offset)	Specifies the bottom edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
l (Left Offset)	Specifies the left edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
r (Right Offset)	Specifies the right edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
t (Top Offset)	Specifies the top edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RelativeRect) is located in §A.4.1. *end note*]

20.1.8.56 stretch (Stretch)

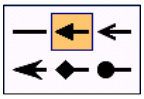
This element specifies that a BLIP should be stretched to fill the target rectangle. The other option is a tile where a BLIP is tiled to fill the available area.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StretchInfoProperties) is located in §A.4.1. *end note*]

20.1.8.57 tailEnd (Tail line end style)

This element specifies decorations which can be added to the tail of a line.

[*Example*:



end example]

Attributes	Description
len (Length of Head/End)	Specifies the line end length in relation to the line width. The possible values for this attribute are defined by the ST_LineEndLength simple type (§20.1.10.32).
type (Line Head/End Type)	Specifies the line end decoration, such as a triangle or arrowhead. The possible values for this attribute are defined by the ST_LineEndType simple type (§20.1.10.33).
w (Width of Head/End)	Specifies the line end width in relation to the line width. The possible values for this attribute are defined by the ST_LineEndWidth simple type (§20.1.10.34).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineEndProperties) is located in §A.4.1. *end note*]

20.1.8.58 tile (Tile)

This element specifies that a BLIP should be tiled to fill the available space. This element defines a "tile" rectangle within the bounding box. The image is encompassed within the tile rectangle, and the tile rectangle is tiled across the bounding box to fill the entire area.

[*Example*:

The following is a fill with the flip attribute set to "x". The black interior rectangle indicates the tile rectangle. Notice that the adjacent rectangle to the right in the tile has been flipped along the x-axis.

end example]

Attributes	Description
algn (Alignment)	Specifies where to align the first tile with respect to the shape. Alignment happens after the scaling, but before the additional offset. The possible values for this attribute are defined by the ST_RectAlignment simple type (§20.1.10.52).
flip (Tile Flipping)	Specifies the direction(s) in which to flip the source image while tiling. Images can be flipped horizontally, vertically, or in both directions to fill the entire region. The possible values for this attribute are defined by the ST_TileFlipMode simple type (§20.1.10.85).
sx (Horizontal Ratio)	Specifies the amount to horizontally scale the srcRect. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
sy (Vertical Ratio)	Specifies the amount to vertically scale the srcRect. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
tx (Horizontal Offset)	Specifies additional horizontal offset after alignment. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
ty (Vertical Offset)	Specifies additional vertical offset after alignment. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TileInfoProperties) is located in §A.4.1. *end note*]

20.1.8.59 tileRect (Tile Rectangle)

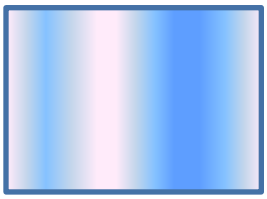
This element specifies a rectangular region of the shape to which the gradient is applied. This region is then tiled across the remaining area of the shape to complete the fill. The tile rectangle is defined by percentage offsets from the sides of the shape's bounding box.

Each edge of the tile rectangle is defined by a percentage offset from the corresponding edge of the bounding box. A positive percentage specifies an inset, while a negative percentage specifies an outset. [*Note*: For example, a left offset of 25% specifies that the left edge of the tile rectangle is located to the right of the bounding box's left edge by an amount equal to 25% of the bounding box's width. *end note*]

[*Example*:

The image above depicts a horizontal gradient with no tileRect element.

The image above depicts the same gradient with a tileRect element specifying l="50000" (50%). The right half of the shape is the tile to which the gradient is applied, and the left half of the shape contains a tiled copy of that gradient fill.



The image above depicts the same gradient with a tileRect element specifying l="75000" (75%). The rightmost 25% of the shape contains the tile rectangle to which the gradient is applied. This gradient is tiled three times to cover the leftmost 75% of the shape. The tile rectangle is flipped horizontally when covering the shape.

end example]

Attributes	Description
b (Bottom Offset)	Specifies the bottom edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type

Attributes	Description
	(§20.1.10.40).
l (Left Offset)	Specifies the left edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
r (Right Offset)	Specifies the right edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
t (Top Offset)	Specifies the top edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RelativeRect) is located in §A.4.1. *end note*]

20.1.8.60 tint (Tint Effect)

This element specifies a tint effect. Shifts effect color values towards/away from hue by the specified amount.

Attributes	Description
amt (Amount)	Specifies by how much the color value is shifted. The possible values for this attribute are defined by the ST_FixedPercentage simple type (§20.1.10.24).
hue (Hue)	Specifies the hue towards which to tint. The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.43).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TintEffect) is located in §A.4.1. *end note*]

20.1.8.61 xfrm (Transform Effect)

This element specifies a transform effect. The transform is applied to each point in the shape's geometry using the following matrix:

$$\begin{bmatrix} sx & \tan(kx) & tx \\ \tan(ky) & sy & ty \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Attributes	Description
kx (Horizontal Skew)	Specifies the horizontal skew angle, defined as the angle between the top-left corner and bottom-left corner of the object's original bounding box. If positive, the bottom edge of the shape is positioned to the right relative to the top edge. The possible values for this attribute are defined by the ST_FixedAngle simple type (§20.1.10.23).
ky (Vertical Skew)	Specifies the vertical skew angle, defined as the angle between the top-left corner and top-right corner of the object's original bounding box. If positive, the right edge of the object is positioned lower relative to the left edge. The possible values for this attribute are defined by the ST_FixedAngle simple type (§20.1.10.23).
sx (Horizontal Ratio)	Specifies a percentage by which to horizontally scale the object. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
sy (Vertical Ratio)	Specifies a percentage by which to vertically scale the object. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
tx (Horizontal Shift)	Specifies an amount by which to shift the object along the x-axis. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
ty (Vertical Shift)	Specifies an amount by which to shift the object along the y-axis. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TransformEffect) is located in §A.4.1. *end note*]

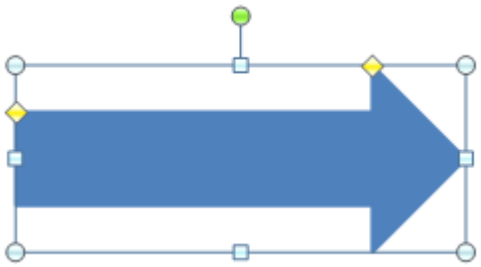
20.1.9 Shape Definitions and Attributes

The Shape Definitions and Attributes portion of the DrawingML framework deals with all geometric properties for shapes within a document. This includes both preset geometries that publicly are interpreted by the generating application and custom geometries that have their points and curves explicitly specified. In addition to the underlying geometry of the shape there are also other coordinate-based properties for each shape that this framework describes.

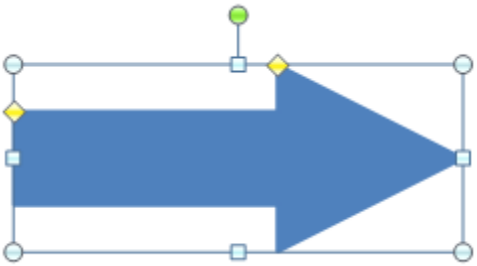
20.1.9.1 ahLst (List of Shape Adjust Handles)

This element specifies the adjust handles that are applied to a custom geometry. These adjust handles specify points within the geometric shape that can be used to perform certain transform operations on the shape.

[*Example*: Consider the scenario where a custom geometry, an arrow in this case, has been drawn and adjust handles have been placed at the top left corner of both the arrow head and arrow body. The user interface can then be made to transform only certain parts of the shape by using the corresponding adjust handle.



For instance if the user wished to change only the width of the arrow head then they would use the adjust handle located on the top left of the arrow head. The result of adjusting this transforms the shape as shown below.



end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_AdjustHandleList) is located in §A.4.1. *end note*]

20.1.9.2 ahPolar (Polar Adjust Handle)

This element specifies a polar adjust handle for a custom shape. The position of this adjust handle is specified by the corresponding pos child element. The allowed adjustment of this adjust handle are specified via it's min and max attributes. Based on the adjustment of this adjust handle certain corresponding guides are updated to contain these values.

Attributes	Description
gdRefAng (Angle Adjustment Guide)	Specifies the name of the guide that is updated with the adjustment angle from this adjust handle.

Attributes	Description
	The possible values for this attribute are defined by the ST_GeomGuideName simple type (§20.1.10.28).
gdRefR (Radial Adjustment Guide)	Specifies the name of the guide that is updated with the adjustment radius from this adjust handle. The possible values for this attribute are defined by the ST_GeomGuideName simple type (§20.1.10.28).
maxAng (Maximum Angle Adjustment)	Specifies the maximum angle position that is allowed for this adjustment handle. If this attribute is omitted, then it is assumed that this adjust handle cannot move angularly. That is the maxAng and minAng are equal. The possible values for this attribute are defined by the ST_AdjAngle simple type (§20.1.10.1).
maxR (Maximum Radial Adjustment)	Specifies the maximum radial position that is allowed for this adjustment handle. If this attribute is omitted, then it is assumed that this adjust handle cannot move radially. That is the maxR and minR are equal. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).
minAng (Minimum Angle Adjustment)	Specifies the minimum angle position that is allowed for this adjustment handle. If this attribute is omitted, then it is assumed that this adjust handle cannot move angularly. That is the maxAng and minAng are equal. The possible values for this attribute are defined by the ST_AdjAngle simple type (§20.1.10.1).
minR (Minimum Radial Adjustment)	Specifies the minimum radial position that is allowed for this adjustment handle. If this attribute is omitted, then it is assumed that this adjust handle cannot move radially. That is the maxR and minR are equal. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PolarAdjustHandle) is located in §A.4.1. *end note*]

20.1.9.3 ahXY (XY Adjust Handle)

This element specifies an XY-based adjust handle for a custom shape. The position of this adjust handle is specified by the corresponding pos child element. The allowed adjustment of this adjust handle are specified via it's min and max type attributes. Based on the adjustment of this adjust handle certain corresponding guides are updated to contain these values.

Attributes	Description
gdRefX (Horizontal Adjustment Guide)	Specifies the name of the guide that is updated with the adjustment x position from this adjust handle. The possible values for this attribute are defined by the ST_GeomGuideName simple type (§20.1.10.28).
gdRefY (Vertical Adjustment Guide)	Specifies the name of the guide that is updated with the adjustment y position from this adjust handle. The possible values for this attribute are defined by the ST_GeomGuideName simple type (§20.1.10.28).
maxX (Maximum Horizontal Adjustment)	Specifies the maximum horizontal position that is allowed for this adjustment handle. If this attribute is omitted, then it is assumed that this adjust handle cannot move in the x direction. That is the maxX and minX are equal. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).
maxY (Maximum Vertical Adjustment)	Specifies the maximum vertical position that is allowed for this adjustment handle. If this attribute is omitted, then it is assumed that this adjust handle cannot move in the y direction. That is the maxY and minY are equal. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).
minX (Minimum Horizontal Adjustment)	Specifies the minimum horizontal position that is allowed for this adjustment handle. If this attribute is omitted, then it is assumed that this adjust handle cannot move in the x direction. That is the maxX and minX are equal. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).
minY (Minimum Vertical Adjustment)	Specifies the minimum vertical position that is allowed for this adjustment handle. If this attribute is omitted, then it is assumed that this adjust handle cannot move in the y direction. That is the maxY and minY are equal. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).

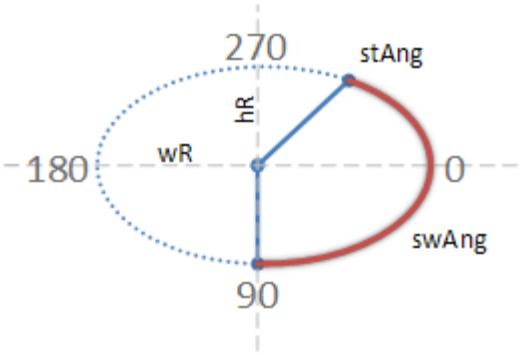
[*Note*: The W3C XML Schema definition of this element's content model (CT_XYAdjustHandle) is located in §A.4.1. *end note*]

20.1.9.4 arcTo (Draw Arc To)

This element specifies the existence of an arc within a shape path. It draws an arc with the specified parameters from the current pen position to the new point specified. An arc is a line that is bent based on the shape of a

supposed circle. The length of this arc is determined by specifying both a start angle and an ending angle that act together to effectively specify an end point for the arc.

[*Example*: The diagram shown below represents a single arc that has a start angle of 300 degrees and a swing angle of 150 degrees. This arc is drawn using the supposed circle that is described using the hR and wR attributes as shown below. The degrees by which the stAng must abide is shown along the circumference of the circle. These degrees are to be specified in 60,000ths of a degree. If this arc were part of a shape the start angle point along the circle would be the starting point along the path and the ending point would be the ending of the angle swing along this supposed circle. That is any shape geometry coming before this arc in the shape path would be joined with the upper point of this arc and consequently any geometry coming after this arc in the path would be joined with the lower point of this arc.



end example]

Attributes	Description
hR (Shape Arc Height Radius)	This attribute specifies the height radius of the supposed circle being used to draw the arc. This gives the circle a total height of (2 * hR). This total height could also be called it's vertical diameter as it is the diameter for the y axis only. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).
stAng (Shape Arc Start Angle)	Specifies the start angle for an arc. This angle specifies what angle along the supposed circle path is used as the start position for drawing the arc. This start angle is locked to the last known pen position in the shape path. Thus guaranteeing a continuos shape path. The possible values for this attribute are defined by the ST_AdjAngle simple type (§20.1.10.1).
swAng (Shape Arc Swing Angle)	Specifies the swing angle for an arc. This angle specifies how far angle-wise along the supposed cicle path the arc is extended. The extension from the start angle is always in the clockwise direction around the supposed circle. The possible values for this attribute are defined by the ST_AdjAngle simple type (§20.1.10.1).

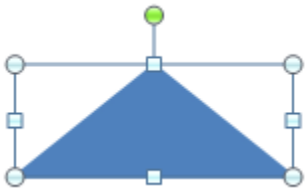
Attributes	Description
wR (Shape Arc Width Radius)	This attribute specifies the width radius of the supposed circle being used to draw the arc. This gives the circle a total width of (2 * wR). This total width could also be called it's horizontal diameter as it is the diameter for the x axis only. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Path2DArcTo) is located in §A.4.1. *end note*]

20.1.9.5 avLst (List of Shape Adjust Values)

This element specifies the adjust values that are applied to the specified shape. An adjust value is simply a guide that has a value based formula specified. That is, no calculation takes place for an adjust value guide. Instead, this guide specifies a parameter value that is used for calculations within the shape guides.

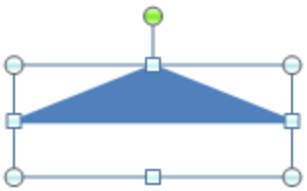
[*Example*: Consider the case where the user would like to specify a triangle with it's bottom edge defined not by static points but by using a varying parameter, namely an adjust value. Consider the diagrams and DrawingML shown below. This first triangle has been drawn with a bottom edge that is equal to the height, namely 2. Thus we see in the figure below that the bottom of the triangle matches the bottom of the shape bounding box.



```
<a:xfrm>
  <a:off x="3200400" y="1600200"/>
  <a:ext cx="1705233" cy="679622"/>
</a:xfrm>
<a:custGeom>
  <a:avLst>
    <a:gd name="myGuide" fmla="val 2"/>
  </a:avLst>
  <a:gdLst/>
  <a:ahLst/>
  <a:cxnLst/>
  <a:rect l="0" t="0" r="0" b="0"/>
```

```
    <a:pathLst>
      <a:path w="2" h="2">
        <a:moveTo>
          <a:pt x="0" y="myGuide"/>
        </a:moveTo>
        <a:lnTo>
          <a:pt x="2" y="myGuide"/>
        </a:lnTo>
        <a:lnTo>
          <a:pt x="1" y="0"/>
        </a:lnTo>
        <a:close/>
      </a:path>
    </a:pathLst>
  </a:custGeom>
```

If however we change the adjust value to half that, namely 1. Then we see the entire bottom edge of the triangle move to now be placed along the vertical midpoint within the shape bounding box. This is because both of the bottom points in this triangle depend on this adjust value for their coordinate positions. The triangle and corresponding DrawingML shown below illustrate this point.



```
  <a:avLst>
    <a:gd name="myGuide" fmla="val 1"/>
  </a:avLst>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GeomGuideList) is located in §A.4.1. *end note*]

20.1.9.6 close (Close Shape Path)

This element specifies the ending of a series of lines and curves in the creation path of a custom geometric shape. When this element is encountered, the generating application should consider the corresponding path closed. That is, any further lines or curves that follow this element should be ignored.

[*Note*: A path can be specified and not closed. A path such as this cannot however have any fill associated with it as it has not been considered a closed geometric path. *end note*]

[*Example*: Consider the following DrawingML.

```
<a:custGeom>
  <a:pathLst>
    <a:path w="2824222" h="590309">
      <a:moveTo>
        <a:pt x="0" y="428263"/>
      </a:moveTo>
      <a:lnTo>
        <a:pt x="1620455" y="590309"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="2824222" y="173620"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="1562582" y="0"/>
      </a:lnTo>
      <a:close/>
    </a:path>
  </a:pathLst>
</a:custGeom>
```

In the above example there is specified a four sided geometric shape that has all straight sides. While we only see three lines being drawn via the lnTo element there are actually four sides because the last point of (x=1562585, y=0) is connected to the first point in the creation path via a lnTo element. *end example*]

[*Note*: When the last point in the creation path does not meet with the first point in the creation path the generating application should connect the last point with the first via a straight line, thus creating a closed shape geometry. *end note*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Path2DClose) is located in §A.4.1. *end note*]

20.1.9.7 cubicBezTo (Draw Cubic Bezier Curve To)

This element specifies to draw a cubic bezier curve along the specified points. To specify a cubic bezier curve there needs to be 3 points specified. The first two are control points used in the cubic bezier calculation and the last is the ending point for the curve. The coordinate system used for this kind of curve is the path coordinate system as this element is path specific.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Path2DCubicBezierTo) is located in §A.4.1. *end note*]

20.1.9.8 custGeom (Custom Geometry)

This element specifies the existence of a custom geometric shape. This shape consists of a series of lines and curves described within a creation path. In addition to this there can also be adjust values, guides, adjust handles, connection sites and an inscribed rectangle specified for this custom geometric shape.

[*Example*: Consider the scenario when a preset geometry does not accurately depict what must be displayed in the document. For this a custom geometry can be used to define most any 2-dimensional geometric shape. Shown below is an example of such a custom geometry.

```
<a:custGeom>
  <a:avLst/>
  <a:gdLst/>
  <a:ahLst/>
  <a:cxnLst/>
  <a:rect l="0" t="0" r="0" b="0"/>
  <a:pathLst>
    <a:path w="2650602" h="1261641">
      <a:moveTo>
        <a:pt x="0" y="1261641"/>
      </a:moveTo>
      <a:lnTo>
        <a:pt x="2650602" y="1261641"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="1226916" y="0"/>
      </a:lnTo>
      <a:close/>
    </a:path>
  </a:pathLst>
</a:custGeom>
```

The custom geometry above is drawn by first moving to a specific starting point with the moveTo element. Then a series of lnTo elements in the creation path specify the lines that make up the borders of the shape and finally a close element is used to specify the end of the creation path. The resulting shape is shown above. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomGeometry2D) is located in §A.4.1. *end note*]

20.1.9.9 cxn (Shape Connection Site)

This element specifies the existence of a connection site on a custom shape. A connection site allows a cxnSp to be attached to this shape. This connection is maintained when the shape is repositioned within the document. It should be noted that this connection is placed within the shape bounding box using the transform coordinate system which is also called the shape coordinate system, as it encompasses the entire shape. The width and height for this coordinate system are specified within the ext transform element.

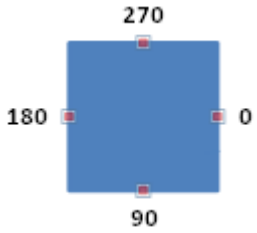
[*Note*: The transform coordinate system is different from a path coordinate system as it is per shape instead of per path within the shape. *end note*]

[*Example*: Consider the following custom geometry that has two connection sites specified. One connection is located at the bottom left of the shape and the other at the bottom right. The following DrawingML would describe such a custom geometry.

```
<a:xfrm>
  <a:off x="3200400" y="1600200"/>
  <a:ext cx="1705233" cy="679622"/>
</a:xfrm>
<a:custGeom>
  <a:avLst/>
  <a:gdLst/>
  <a:ahLst/>
  <a:cxnLst>
    <a:cxn ang="0">
      <a:pos x="0" y="679622"/>
    </a:cxn>
    <a:cxn ang="0">
      <a:pos x="1705233" y="679622"/>
    </a:cxn>
  </a:cxnLst>
```

```
    <a:rect l="0" t="0" r="0" b="0"/>
    <a:pathLst>
      <a:path w="2" h="2">
        <a:moveTo>
          <a:pt x="0" y="2"/>
        </a:moveTo>
        <a:lnTo>
          <a:pt x="2" y="2"/>
        </a:lnTo>
        <a:lnTo>
          <a:pt x="1" y="0"/>
        </a:lnTo>
        <a:close/>
      </a:path>
    </a:pathLst>
  </a:custGeom>
```

end example]

Attributes	Description
ang (Connection Site Angle)	Specifies the incoming connector angle. This angle is the angle around the connection site that an incoming connector tries to be routed to. This allows connectors to know where the shape is in relation to the connection site and route connectors so as to avoid any overlap with the shape. [*Example*: Consider a simple square. In order to not have any connectors routed over the shape, the collowing angles would be specified for their respective connection sites.  *end example*] The possible values for this attribute are defined by the ST_AdjAngle simple type (§20.1.10.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ConnectionSite) is located in §A.4.1. *end note*]

20.1.9.10 cxnLst (List of Shape Connection Sites)

This element specifies all the connection sites that are used for this shape. A connection site is specified by defining a point within the shape bounding box that can have a cxnSp element attached to it. These connection sites are specified using the shape coordinate system that is specified within the ext transform element.

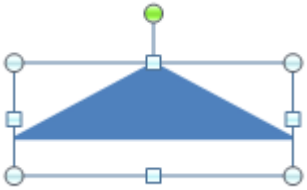
[*Note*: The W3C XML Schema definition of this element's content model (CT_ConnectionSiteList) is located in §A.4.1. *end note*]

20.1.9.11 gd (Shape Guide)

This element specifies the precense of a shape guide that is used to govern the geometry of the specified shape. A shape guide consists of a formula and a name that the result of the formula is assigned to. Recognized formulas are listed with the fmla attribute documentation for this element.

[*Note*: The order in which guides are specified determines the order in which their values are calculated. For instance it is not possible to specify a guide that uses another guides result when that guide has not yet been calculated. *end note*]

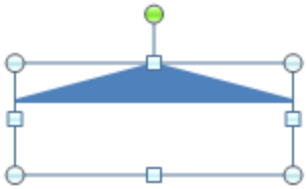
[*Example*: Consider the case where the user would like to specify a triangle with it's bottom edge defined not by static points but by using a varying parameter, namely an guide. Consider the diagrams and DrawingML shown below. This first triangle has been drawn with a bottom edge that is equal to the 2/3 the value of the shape height. Thus we see in the figure below that the triangle appears to occupy 2/3 of the vertical space within the shape bounding box.



```
<a:xfrm>
  <a:off x="3200400" y="1600200"/>
  <a:ext cx="1705233" cy="679622"/>
</a:xfrm>
<a:custGeom>
  <a:avLst/>
  <a:gdLst>
    <a:gd name="myGuide" fmla="*/ h 2 3"/>
  </a:gdLst>
  <a:ahLst/>
  <a:cxnLst/>
  <a:rect l="0" t="0" r="0" b="0"/>
  <a:pathLst>
```

```
      <a:path w="1705233" h="679622">
        <a:moveTo>
          <a:pt x="0" y="myGuide"/>
        </a:moveTo>
        <a:lnTo>
          <a:pt x="1705233" y="myGuide"/>
        </a:lnTo>
        <a:lnTo>
          <a:pt x="852616" y="0"/>
        </a:lnTo>
        <a:close/>
      </a:path>
    </a:pathLst>
  </a:custGeom>
```

If however we change the guide to half that, namely 1/3. Then we see the entire bottom edge of the triangle move to now only occupy 1/3 of the toal space within the shape bounding box. This is because both of the bottom points in this triangle depend on this guide for their coordinate positions. The triangle and corresponding DrawingML shown below illustrate this point.



```
  <a:gdLst>
    <a:gd name="myGuide" fmla="*/ h 1 3"/>
  </a:gdLst>
```

end example]

Attributes	Description
fmla (Shape Guide Formula)	Specifies the formula that is used to calculate the value for a guide. Each formula has a certain number of arguments and a specific set of operations to perform on these arguments in order to generate a value for a guide. There are a total of 17 different formulas available. These are shown below with the usage for each defined. **('*/') - Multiply Divide Formula** Arguments: 3 (fmla="*/ x y z") Usage: "*/ x y z" = ((x * y) / z) = value of this guide **('+-') - Add Subtract Formula** Arguments: 3 (fmla="+- x y z") Usage: "+- x y z" = ((x + y) - z) = value of this guide

Attributes	Description
	('+/') - Add Divide Formula Arguments: 3 (fmla="+/ x y z") Usage: "+/ x y z" = ((x + y) / z) = value of this guide **('?:') - If Else Formula** Arguments: 3 (fmla="?: x y z") Usage: "?: x y z" = if (x > 0), then y = value of this guide, else z = value of this guide **('abs') - Absolute Value Formula** Arguments: 1 (fmla="abs x") Usage: "abs x" = if (x < 0), then (-1) * x = value of this guide else x = value of this guide **('at2') - ArcTan Formula** Arguments: 2 (fmla="at2 x y") Usage: "at2 x y" = arctan(y / x) = value of this guide **('cat2') - Cosine ArcTan Formula** Arguments: 3 (fmla="cat2 x y z") Usage: "cat2 x y z" = (x*(cos(arctan(z / y))) = value of this guide **('cos') - Cosine Formula** Arguments: 2 (fmla="cos x y") Usage: "cos x y" = (x * cos(y)) = value of this guide **('max') - Maximum Value Formula** Arguments: 2 (fmla="max x y") Usage: "max x y" = if (x > y), then x = value of this guide else y = value of this guide **('min') - Minimum Value Formula** Arguments: 2 (fmla="min x y") Usage: "min x y" = if (x < y), then x = value of this guide else y = value of this guide **('mod') - Modulo Formula** Arguments: 3 (fmla="mod x y z") Usage: "mod x y z" = sqrt(x^2 + b^2 + c^2) = value of this guide **('pin') - Pin To Formula** Arguments: 3 (fmla="pin x y z") Usage: "pin x y z" = if (y < x), then x = value of this guide else if (y > z), then z = value of this guide else y = value of this guide **('sat2') - Sine ArcTan Formula** Arguments: 3 (fmla="sat2 x y z") Usage: "sat2 x y z" = (x*sin(arctan(z / y))) = value of this guide **('sin') - Sine Formula** Arguments: 2 (fmla="sin x y")

Attributes	Description
	Usage: "sin x y" = (x * sin(y)) = value of this guide **('sqrt') - Square Root Formula** Arguments: 1 (fmla="sqrt x") Usage: "sqrt x" = sqrt(x) = value of this guide **('tan') - Tangent Formula** Arguments: 2 (fmla="tan x y") Usage: "tan x y" = (x * tan(y)) = value of this guide **('val') - Literal Value Formula** Arguments: 1 (fmla="val x") Usage: "val x" = x = value of this guide [*Note*: Guides that have a literal value formula specified via fmla="val x" above should only be used within the avLst as an adjust value for the shape. This however is not strictly enforced. *end note*] The possible values for this attribute are defined by the ST_GeomGuideFormula simple type (§20.1.10.27).
name (Shape Guide Name)	Specifies the name that is used to reference to this guide. This name can be used just as a variable would within an equation. That is this name can be substituted for literal values within other guides or the specification of the shape path. The possible values for this attribute are defined by the ST_GeomGuideName simple type (§20.1.10.28).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GeomGuide) is located in §A.4.1. *end note*]

20.1.9.12 gdLst (List of Shape Guides)

This element specifies all the guides that are used for this shape. A guide is specified by the gd element and defines a calculated value that can be used for the construction of the corresponding shape.

[*Note*: Guides that have a literal value formula specified via fmla="val x" above should only be used within the avLst as an adjust value for the shape. This however is not strictly enforced. *end note*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GeomGuideList) is located in §A.4.1. *end note*]

20.1.9.13 lnTo (Draw Line To)

This element specifies the drawing of a straight line from the current pen position to the new point specified. This line becomes part of the shape geometry, representing a side of the shape. The coordinate system used when specifying this line is the path coordinate system.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Path2DLineTo) is located in §A.4.1. *end note*]

20.1.9.14 moveTo (Move Path To)

This element specifies a set of new coordinates to move the shape cursor to. This element is only used for drawing a custom geometry. When this element is utilized the pt element is used to specify a new set of shape coordinates that the shape cursor should be moved to. This does not draw a line or curve to this new position from the old position but simply move the cursor to a new starting position. It is only when a path drawing element such as lnTo is used that a portion of the path is drawn.

[*Example*: Consider the case where a user wishes to begin drawing a custom geometry not at the default starting coordinates of x=0 , y=0 but at coordinates further inset into the shape coordinate space. The following DrawingML would specify such a case.

```
<a:custGeom>
  <a:pathLst>
    <a:path w="2824222" h="590309">
      <a:moveTo>
        <a:pt x="0" y="428263"/>
      </a:moveTo>
      <a:lnTo>
        <a:pt x="1620455" y="590309"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="2824222" y="173620"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="1562582" y="0"/>
      </a:lnTo>
      <a:close/>
    </a:path>
  </a:pathLst>
</a:custGeom>
```

Notice the moveTo element advances the y coordinates before any actual lines are drawn. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Path2DMoveTo) is located in §A.4.1. *end note*]

20.1.9.15 path (Shape Path)

This element specifies a creation path consisting of a series of moves, lines and curves that when combined forms a geometric shape. This element is only utilized if a custom geometry is specified.

[*Note*: Since multiple paths are allowed the rules for drawing are that the path specified later in the pathLst is drawn on top of all previous paths. *end note*]

[*Example*: Consider the following DrawingML.

```
<a:custGeom>
  <a:pathLst>
    <a:path w="2824222" h="590309">
      <a:moveTo>
        <a:pt x="0" y="428263"/>
      </a:moveTo>
      <a:lnTo>
        <a:pt x="1620455" y="590309"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="2824222" y="173620"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="1562582" y="0"/>
      </a:lnTo>
      <a:close/>
    </a:path>
  </a:pathLst>
</a:custGeom>
```

In the above example there is specified a four sided geometric shape that has all straight sides. While we only see three lines being drawn via the lnTo element there are actually four sides because the last point of (x=1562585, y=0) is connected to the first point in the creation path via a lnTo element. *end example*]

Attributes	Description
extrusionOk (3D Extrusion Allowed)	Specifies that the use of 3D extrusions are possible on this path. This allows the generating application to know whether 3D extrusion can be applied in any form. If this attribute is omitted then a value of 0, or false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fill (Path Fill)	Specifies how the corresponding path should be filled. If this attribute is omitted, a value of "norm" is assumed. The possible values for this attribute are defined by the ST_PathFillMode simple type (§20.1.10.37).
h (Path Height)	Specifies the height, or maximum y coordinate that should be used for within the path coordinate system. This value determines the vertical placement of all points within the corresponding path as they are all calculated using this height attribute as the max y

Attributes	Description
	coordinate. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
stroke (Path Stroke)	Specifies if the corresponding path should have a path stroke shown. This is a boolean value that affect the outline of the path. If this attribute is omitted, a value of true is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
w (Path Width)	Specifies the width, or maximum x coordinate that should be used for within the path coordinate system. This value determines the horizontal placement of all points within the corresponding path as they are all calculated using this width attribute as the max x coordinate. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Path2D) is located in §A.4.1. *end note*]

20.1.9.16 pathLst (List of Shape Paths)

This element specifies the entire path that is to make up a single geometric shape. The pathLst can consist of many individual paths within it.

[*Example*: Consider the following DrawingML.

```
<a:custGeom>
  <a:pathLst>
    <a:path w="2824222" h="590309">
      <a:moveTo>
        <a:pt x="0" y="428263"/>
      </a:moveTo>
      <a:lnTo>
        <a:pt x="1620455" y="590309"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="2824222" y="173620"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="1562582" y="0"/>
      </a:lnTo>
```

```
        <a:close/>
      </a:path>
    </a:pathLst>
  </a:custGeom>
```

In the above example there is specified a four sided geometric shape that has all straight sides. While we only see three lines being drawn via the lnTo element there are actually four sides because the last point of (x=1562585, y=0) is connected to the first point in the creation path via a lnTo element. *end example*]

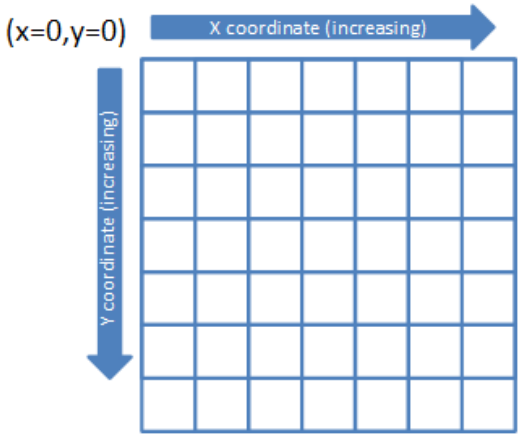
[*Note*: A geometry with multiple paths within it should be treated visually as if each path were a distinct shape. That is each creation path has its first point and last point joined to form a closed shape. However, the generating application should then connect the last point to the first point of the new shape. If a close element is encountered at the end of the previous creation path then this joining line should not be rendered by the generating application. The rendering should resume with the first line or curve on the new creation path. *end note*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Path2DList) is located in §A.4.1. *end note*]

20.1.9.17 pos (Shape Position Coordinate)

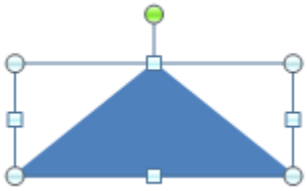
Specifies a position coordinate within the shape bounding box. It should be noted that this coordinate is placed within the shape bounding box using the transform coordinate system which is also called the shape coordinate system, as it encompasses the entire shape. The width and height for this coordinate system are specified within the ext transform element.

[*Note*: When specifying a point coordinate in path coordinate space it should be noted that the top left of the coordinate space is x=0, y=0 and the coordinate points for x grow to the right and for y grow down. This is illustrated in the diagram below.



end note]

[*Example*: To highlight the differences in the coordinate systems consider the drawing of the following triangle. Notice that the dimensions of the triangle are specified using the shape coordinate system with EMUs as the units via the ext transform element. Thus we see this shape is 1705233 EMUs wide by 679622 EMUs tall. However when looking at how the path for this shape is drawn we see that the x and y values fall between 0 and 2. This is because the path coordinate system has the arbitrary dimensions of 2 for the width and 2 for the height. Thus we see that a y coordinate of 2 within the path coordinate system specifies a y coordinate of 679622 within the shape coordinate system for this particular case.



```
<a:xfrm>
  <a:off x="3200400" y="1600200"/>
  <a:ext cx="1705233" cy="679622"/>
</a:xfrm>
<a:custGeom>
  <a:avLst/>
    <a:gdLst/>
  <a:ahLst/>
  <a:cxnLst/>
  <a:rect l="0" t="0" r="0" b="0"/>
  <a:pathLst>
    <a:path w="2" h="2">
      <a:moveTo>
        <a:pt x="0" y="2"/>
      </a:moveTo>
      <a:lnTo>
        <a:pt x="2" y="2"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="1" y="0"/>
      </a:lnTo>
      <a:close/>
    </a:path>
  </a:pathLst>
</a:custGeom>
```

end example]

Attributes	Description
x (X-Coordinate)	Specifies the x coordinate for this position coordinate. The units for this coordinate space are defined by the width of the path coordinate system. This coordinate system is overlayed on top of the shape coordinate system thus occupying the entire shape bounding box. Because the units for within this coordinate space are determined by the path width and height an exact measurement unit cannot be specified here. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).
y (Y-Coordinate)	Specifies the y coordinate for this position coordinate. The units for this coordinate space are defined by the height of the path coordinate system. This coordinate system is overlayed on top of the shape coordinate system thus occupying the entire shape bounding box. Because the units for within this coordinate space are determined by the path width and height an exact measurement unit cannot be specified here. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AdjPoint2D) is located in §A.4.1. *end note*]

20.1.9.18 prstGeom (Preset geometry)

This element specifies when a preset geometric shape should be used instead of a custom geometric shape. The generating application should be able to render all preset geometries enumerated in the ST_ShapeType list.

[*Example*: Consider the scenario when a user does not wish to specify all the lines and curves that make up the desired shape but instead chooses to use a preset geometry. The following DrawingML would specify such a case.

```
<p:sp>
  <p:nvSpPr>
    <p:cNvPr id="4" name="My Preset Shape"/>
    <p:cNvSpPr/>
    <p:nvPr/>
  </p:nvSpPr>
  <p:spPr>
    <a:xfrm>
      <a:off x="1981200" y="533400"/>
      <a:ext cx="1143000" cy="1066800"/>
    </a:xfrm>
    <a:prstGeom prst="heart">
    </a:prstGeom>
  </p:spPr>
</p:sp>
```


The output shape rendered by this DrawingML is shown above. *end example*]

Attributes	Description

prst (Preset Shape)	Specifies the preset geometry that is used for this shape. This preset can have any of the values in the enumerated list for ST_ShapeType. This attribute is required in order for a preset geometry to be rendered.
	
	[*Example*: Consider the sample DrawingML below.
	
	`<p:sp>`
	`  <p:nvSpPr>`
	`  <p:cNvPr id="4" name="Sun 3"/>`
	`  <p:cNvSpPr/>`
	`  <p:nvPr/>`
	`  </p:nvSpPr>`
	`  <p:spPr>`
	`  <a:xfrm>`
	`  <a:off x="1981200" y="533400"/>`
	`  <a:ext cx="1143000" cy="1066800"/>`
	`  </a:xfrm>`
	`  <a:prstGeom prst="sun">`
	`  </a:prstGeom>`
	`  </p:spPr>`
	`</p:sp>`
	
	In the above example a preset geometry has been used to define a shape. The shape utilized here is the sun shape. *end example*]
	
	The possible values for this attribute are defined by the ST_ShapeType simple type (§20.1.10.55).

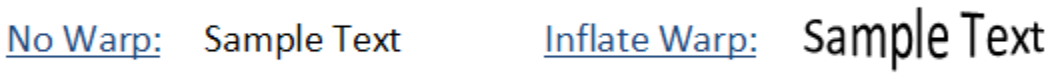
[*Note*: The W3C XML Schema definition of this element's content model (CT_PresetGeometry2D) is located in §A.4.1. *end note*]

20.1.9.19 prstTxWarp (Preset Text Warp)

This element specifies when a preset geometric shape should be used to transform a piece of text. This operation is known formally as a text warp. The generating application should be able to render all preset geometries enumerated in the ST_TextShapeType list.

[*Example*: Consider the case where the user wishes to accent a piece of text by warping it's shape. For this to occur a preset shape is chosen from the ST_TextShapeType list and applied to the entire body of text.

```
<p:sp>
  <p:txBody>
    <a:bodyPr wrap="none" rtlCol="0">
      <a:prstTxWarp prst="textInflate">
      </a:prstTxWarp>
      <a:spAutoFit/>
    </a:bodyPr>
    <a:lstStyle/>
    <a:p>
…
      <a:t>Sample Text</a:t>
…
    </a:p>
  </p:txBody>
</p:sp>
```



The resulting text that has now had the Inflate text warp applied to it is shown above. *end example*]

Using any of the presets listed under the ST_TextShapeType list below it is possible to apply a text warp to a run of DrawingML text via the following steps.

If you look at any of the text warps in the file format you notice that each consists of two paths. This corresponds to a top path (first one specified) and a bottom path (second one specified). Now the top path and the bottom path represent the top line and base line that the text needs to be warped to. This is done in the following way:

1. Compute the rectangle that the unwarped text resides in. (tightest possible rectangle around text, no white space except for “space characters”)
2. Take each of the quadratic and cubic Bezier curves that are used to calculate the original character and change their end points and control points by the following method…
3. Move a vertical line horizontally along the original text rectangle and find the horizontal percentage that a given end point or control point lives at. (.5 for the middle for instance)
4. Now do the same thing for this point vertically. Find the vertical percentage that this point lives at with the top and bottom of this text rectangle being the respective top and bottom bounds. (0.0 and 1.0 respectively)

5. Now that we have the percentages for a given point in a Bezier equation we can map that to the new point in the warped text environment.
6. Going back to the top and bottom paths specified in the file format we can take these and flatten them out to a straight arc (top and bottom might be different lengths)
7. After they are straight we can measure them both horizontally to find the same percentage point that we found within the original text rectangle. (0.5 let's say)
8. So then we measure 50% along the top path and 50% along the bottom path, putting the paths back to their original curvy shapes.
9. Once we have these two points we can draw a line between them that serves as our vertical line in the original text rectangle [*Note*: This might not be truly vertical as 50% on the top does not always line up with 50% on the bottom. *end note*]
10. Taking this new line we then follow it from top to bottom the vertical percentage amount that we got from step 4.
11. This is then the new point that should be used in place of the old point in the original text rectangle.
12. We then continue doing these same steps for each of the end points and control points within the body of text. (is applied to a whole body of text only)

[*Note*: Horizxontal percentages begin at 0.0 and continue to 1.0, left to right. Vertical percentages begin at 0.0 and continue to 1.0, top to bottom. *end note*]

[*Note*: Since this is a shape it does have both a shape coordinate system and a path coordinate system. *end note*]

Attributes	Description

prst (Preset Warp Shape)	Specifies the preset geometry that is used for a shape warp on a piece of text. This preset can have any of the values in the enumerated list for ST_TextShapeType. This attribute is required in order for a text warp to be rendered.
	
	[*Example*: Consider the sample DrawingML below.
	
	`<p:sp>`
	`  <p:txBody>`
	`  <a:bodyPr wrap="none" rtlCol="0">`
	`  <a:prstTxWarp prst="textInflate">`
	`  </a:prstTxWarp>`
	`  <a:spAutoFit/>`
	`  </a:bodyPr>`
	`  <a:lstStyle/>`
	`  <a:p>`
	`  …`
	`  <a:t>Sample Text</a:t>`
	`  …`
	`  </a:p>`
	`  </p:txBody>`
	`</p:sp>`
	
	In the above example a preset text shape geometry has been used to define the warping

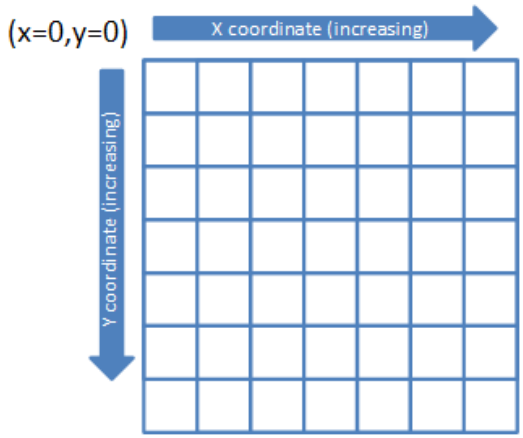
Attributes	Description
	shape. The shape utilized here is the sun shape. *end example*] The possible values for this attribute are defined by the ST_TextShapeType simple type (§20.1.10.75).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PresetTextShape) is located in §A.4.1. *end note*]

20.1.9.20 pt (Shape Path Point)

This element specifies an x-y coordinate within the path coordinate space. This coordinate space is determined by the width and height attributes defined within the path element. A point is utilized by one of it's parent elements to specify the next point of interest in custom geometry shape. Depending on the parent element used the point can either have a line drawn to it or the cursor can simply be moved to this new location.

[*Note*: When specifying a point coordinate in path coordinate space it should be noted that the top left of the coordinate space is x=0, y=0 and the coordinate points for x grow to the right and for y grow down. This is illustrated in the diagram below.

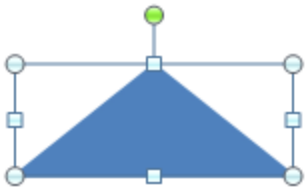


end note]

Specifies a position coordinate within the shape bounding box. It should be noted that this coordinate is placed within the shape bounding box using the transform coordinate system which is also called the shape coordinate system, as it encompasses the entire shape. The width and height for this coordinate system are specified within the ext transform element.

[*Example*: To highlight the differences in the coordinate systems consider the drawing of the following triangle. Notice that the dimensions of the triangle are specified using the shape coordinate system with EMUs as the units via the ext transform element. Thus we see this shape is 1705233 EMUs wide by 679622 EMUs tall. However when looking at how the path for this shape is drawn we see that the x and y values fall between 0 and

2. This is because the path coordinate system has the arbitrary dimensions of 2 for the width and 2 for the height. Thus we see that a y coordinate of 2 within the path coordinate system specifies a y coordinate of 679622 within the shape coordinate system for this particular case.



```
<a:xfrm>
  <a:off x="3200400" y="1600200"/>
  <a:ext cx="1705233" cy="679622"/>
</a:xfrm>
<a:custGeom>
  <a:avLst/>
    <a:gdLst/>
  <a:ahLst/>
  <a:cxnLst/>
  <a:rect l="0" t="0" r="0" b="0"/>
  <a:pathLst>
    <a:path w="2" h="2">
      <a:moveTo>
        <a:pt x="0" y="2"/>
      </a:moveTo>
      <a:lnTo>
        <a:pt x="2" y="2"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="1" y="0"/>
      </a:lnTo>
      <a:close/>
    </a:path>
  </a:pathLst>
</a:custGeom>
```

end example]

Attributes	Description
x (X-Coordinate)	Specifies the x coordinate for this position coordinate. The units for this coordinate space are defined by the width of the path coordinate system. This coordinate system is overlayed on top of the shape coordinate system thus occupying the entire shape bounding box. Because the units for within this coordinate space are determined by the path width and height an exact measurement unit cannot be specified here.

Attributes	Description
	The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).
y (Y-Coordinate)	Specifies the y coordinate for this position coordinate. The units for this coordinate space are defined by the height of the path coordinate system. This coordinate system is overlayed on top of the shape coordinate system thus occupying the entire shape bounding box. Because the units for within this coordinate space are determined by the path width and height an exact measurement unit cannot be specified here. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AdjPoint2D) is located in §A.4.1. *end note*]

20.1.9.21 quadBezTo (Draw Quadratic Bezier Curve To)

This element specifies to draw a quadratic bezier curve along the specified points. To specify a quadratic bezier curve there needs to be 2 points specified. The first is a control point used in the quadratic bezier calculation and the last is the ending point for the curve. The coordinate system used for this type of curve is the path coordinate system as this element is path specific.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Path2DQuadBezierTo) is located in §A.4.1. *end note*]

20.1.9.22 rect (Shape Text Rectangle)

This element specifies the rectangular bounding box for text within a custGeom shape. The default for this rectangle is the bounding box for the shape. This can be modified using this elements four attributes to inset or extend the text bounding box.

[*Note*: Text specified to reside within this shape text rectangle can flow outside this bounding box. Depending on the autofit options within the txBody element the text might not entirely reside within this shape text rectangle. *end note*]

Attributes	Description
b (Bottom Position)	Specifies the y coordinate of the bottom edge for a shape text rectangle. The units for this edge is specified in EMUs as the positioning here is based on the shape coordinate system. The width and height for this coordinate system are specified within the ext transform element. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).
l (Left)	Specifies the x coordinate of the left edge for a shape text rectangle. The units for this

Attributes	Description
	edge is specified in EMUs as the positioning here is based on the shape coordinate system. The width and height for this coordinate system are specified within the ext transform element. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).
r (Right)	Specifies the x coordinate of the right edge for a shape text rectangle. The units for this edge is specified in EMUs as the positioning here is based on the shape coordinate system. The width and height for this coordinate system are specified within the ext transform element. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).
t (Top)	Specifies the y coordinate of the top edge for a shape text rectangle. The units for this edge is specified in EMUs as the positioning here is based on the shape coordinate system. The width and height for this coordinate system are specified within the ext transform element. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GeomRect) is located in §A.4.1. *end note*]

20.1.10 Simple Types

This is the complete list of simple types dedicated to DrawingML framework.

20.1.10.1 ST_AdjAngle (Adjustable Angle Methods)

This simple type is an adjustable angle, either an absolute angle or a reference to a geometry guide. The units for an adjustable angle are 60,000ths of a degree.

This simple type is a union of the following types:

- The ST_Angle simple type (§20.1.10.3).
- The ST_GeomGuideName simple type (§20.1.10.28).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AdjAngle) is located in §A.4.1. *end note*]

20.1.10.2 ST_AdjCoordinate (Adjustable Coordinate Methods)

This simple type is an adjustable coordinate is either an absolute coordinate position or a reference to a geometry guide.

This simple type is a union of the following types:

- The ST_Coordinate simple type (§20.1.10.16).
- The ST_GeomGuideName simple type (§20.1.10.28).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AdjCoordinate) is located in §A.4.1. *end note*]

20.1.10.3 ST_Angle (Angle)

This simple type represents an angle in 60,000ths of a degree. Positive angles are clockwise (i.e., towards the positive y axis); negative angles are counter-clockwise (i.e., towards the negative y axis).

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_Angle) is located in §A.4.1. *end note*]

20.1.10.4 ST_AnimationBuildType (Animation Build Type)

This simple type specifies the ways that an animation can be built, or animated.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
allAtOnce (Animate At Once)	Animate all objects as one.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AnimationBuildType) is located in §A.4.1. *end note*]

20.1.10.5 ST_AnimationChartBuildType (Chart Animation Build Type)

This simple type specifies the ways that a chart animation can be built. That is, it specifies the way in which the objects within the chart should be animated.

This simple type is a union of the following types:

- The ST_AnimationBuildType simple type (§20.1.10.4).
- The ST_AnimationChartOnlyBuildType simple type (§20.1.10.6).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AnimationChartBuildType) is located in §A.4.1. *end note*]

20.1.10.6 ST_AnimationChartOnlyBuildType (Chart only Animation Types)

This simple type specifies the build options available only for animating a chart. These options specify the manner in which the objects within the chart should be grouped and animated.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
category (Catefory)	Animate by each category
categoryEl (Category Element)	Animate by each element within the category
series (Series)	Animate by each series.
seriesEl (Series Element)	Animate by each element within the series

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AnimationChartOnlyBuildType) is located in §A.4.1. *end note*]

20.1.10.7 ST_AnimationDgmBuildType (Diagram Animation Build Type)

This simple type specifies the ways that a diagram animation can be built. That is, it specifies the way in which the objects within the diagram graphical object should be animated.

This simple type is a union of the following types:

- The ST_AnimationBuildType simple type (§20.1.10.4).
- The ST_AnimationDgmOnlyBuildType simple type (§20.1.10.8).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AnimationDgmBuildType) is located in §A.4.1. *end note*]

20.1.10.8 ST_AnimationDgmOnlyBuildType (Diagram only Animation Types)

This simple type specifies the build options available only for animating a diagram. These options specify the manner in which the objects within the chart should be grouped and animated.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
lvlAtOnce (Each Level at Once)	Animate the diagram one level at a time, animating the whole level as one object
lvlOne (Level One-by-One)	Animate the diagram by the elements within a level, animating them one level element at a time.
one (Elements One-by-One)	Animate the diagram by elements. For a tree diagram the animation occurs by branch within the diagram tree.

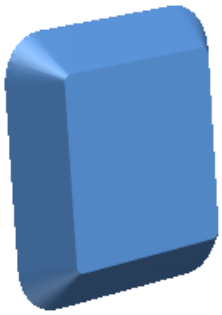
[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AnimationDgmOnlyBuildType) is located in §A.4.1. *end note*]

20.1.10.9 ST_BevelPresetType (Bevel Presets)

Represents a preset for a type of bevel which can be applied to a shape in 3D. The bevel properties are applied differently depending on the type of bevel defined for a shape.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
angle (Angle)	[*Example*: Consider the following example of an `angle` bevel type applied to a shape:  *end example*]
artDeco (Art Deco)	[*Example*: Consider the following example of an `artDeco` bevel type applied to a shape:  *end example*]
circle (Circle)	[*Example*: Consider the following example of an `circle` bevel type applied to a shape:

Enumeration Value	Description
	 *end example*]
convex (Convex)	[*Example*: Consider the following example of an `convex` bevel type applied to a shape:  *end example*]
coolSlant (Cool Slant)	[*Example*: Consider the following example of an `coolSlant` bevel type applied to a shape:  *end example*]
cross (Cross)	[*Example*: Consider the following example of an `cross` bevel type applied to a shape:

Enumeration Value	Description
	 *end example*]
divot (Divot)	[*Example*: Consider the following example of an `divot` bevel type applied to a shape:  *end example*]
hardEdge (Hard Edge)	[*Example*: Consider the following example of an `hardEdge` bevel type applied to a shape:  *end example*]
relaxedInset (Relaxed Inset)	[*Example*: Consider the following example of an

Enumeration Value	Description
	`relaxedInset` bevel type applied to a shape:  *end example*]
riblet (Riblet)	[*Example*: Consider the following example of an `riblet` bevel type applied to a shape:  *end example*]
slope (Slope)	[*Example*: Consider the following example of an `slope` bevel type applied to a shape:  *end example*]

Enumeration Value	Description
softRound (Soft Round)	[*Example*: Consider the following example of an `softRound` bevel type applied to a shape:  *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BevelPresetType) is located in §A.4.1. *end note*]

20.1.10.10 ST_BlackWhiteMode (Black and White Mode)

This simple type specifies how an object should be rendered when specified to be in black and white mode.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
auto (Automatic)	Object rendered with automatic coloring
black (Black)	Object rendered with black-only coloring
blackGray (Black and Gray)	Object rendered with black and gray coloring
blackWhite (Black and White)	Object rendered within black and white coloring
clr (Color)	Object rendered with normal coloring
gray (Gray)	Object rendered with gray coloring
grayWhite (Gray and White)	Object rendered within gray and white coloring
hidden (Hidden)	Object rendered with hidden coloring
invGray (Inverse Gray)	Object rendered with inverse gray coloring
ltGray (Light Gray)	Object rendered with light gray coloring
white (White)	Object rendered within white coloirng

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BlackWhiteMode) is located in §A.4.1. *end note*]

20.1.10.11 ST_BlendMode (Blend Mode)

This simple type describes how to render effects one on top of another.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
darken (Darken)	Darken
lighten (Lighten)	Lighten
mult (Multiply)	Multiply
over (Overlay)	Overlay
screen (Screen)	Screen

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BlendMode) is located in §A.4.1. *end note*]

20.1.10.12 ST_BlipCompression (Blip Compression Type)

This type specifies the amount of compression that has been used for a particular binary large image or picture (blip).

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
email (Email Compression)	Compression size suitable for inclusion with email
hqprint (High Quality Printing Compression)	Compression size suitable for high quality printing
none (No Compression)	No compression was used
print (Printing Compression)	Compression size suitable for printing
screen (Screen Viewing Compression)	Compression size suitable for viewing on screen

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BlipCompression) is located in §A.4.1. *end note*]

20.1.10.13 ST_ChartBuildStep (Chart Animation Build Step)

This simple type specifies an animation build step within a chart animation.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
allPts (All Points)	Animate all points within the chart for this animation build step
category (Category)	Animate a chart category for this animation build step
gridLegend (Grid and Legend)	Animate the chart grid and legend for this animation build step
ptInCategory (Category Points)	Animate a point in a chart category for this animation build step
ptInSeries (Series Points)	Animate a point in a chart series for this animation build step
series (Series)	Animate a chart series for this animation build step

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ChartBuildStep) is located in §A.4.1. *end note*]

20.1.10.14 ST_ColorSchemeIndex (Theme Color Reference)

A reference to a color in the color scheme.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
accent1 (Accent 1)	Represents the accent 1 color.
accent2 (Accent 2)	Represents the accent 2 color.
accent3 (Accent 3)	Represents the accent 3 color.
accent4 (Accent 4)	Represents the accent 4 color.
accent5 (Accent 5)	Represents the accent 5 color.
accent6 (Accent 6)	Represents the accent 6 color.
dk1 (Dark 1)	Represents the first dark color.
dk2 (Dark 2)	Represents the second dark color.
folHlink (Followed Hyperlink)	Represents the followed hyperlink color.
hlink (Hyperlink)	Represents the hyperlink color.
lt1 (Light 1)	Represents the first light color.
lt2 (Light 2)	Represents the second light color.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ColorSchemeIndex) is located in §A.4.1. *end note*]

20.1.10.15 ST_CompoundLine (Compound Line Type)

This simple type specifies the compound line type that is to be used for lines with text such as underlines.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
dbl (Double Lines)	Double lines of equal width
sng (Single Line)	Single line: one normal width
thickThin (Thick Thin Double Lines)	Double lines: one thick, one thin
thinThick (Thin Thick Double Lines)	Double lines: one thin, one thick
tri (Thin Thick Thin Triple Lines)	Three lines: thin, thick, thin

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CompoundLine) is located in §A.4.1. *end note*]

20.1.10.16 ST_Coordinate (Coordinate)

This simple type represents a one dimensional position or length as either:

- EMUs.
- A number followed immediately by a unit identifier.

This simple type is a union of the following types:

- The ST_CoordinateUnqualified simple type (§20.1.10.19).
- The ST_UniversalMeasure simple type (§22.9.2.15).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Coordinate) is located in §A.4.1. *end note*]

20.1.10.17 ST_Coordinate32 (Coordinate Point)

This simple type specifies a coordinate within the document. This can be used for measurements or spacing; its maximum size is 2147483647 EMUs.

Its contents can contain either:

- A whole number, whose contents consist of a measurement in EMUs (English Metric Units)
- A number immediately followed by a unit identifier

This simple type is a union of the following types:

- The ST_Coordinate32Unqualified simple type (§20.1.10.18).
- The ST_UniversalMeasure simple type (§22.9.2.15).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Coordinate32) is located in §A.4.1. *end note*]

20.1.10.18 ST_Coordinate32Unqualified (Coordinate Point)

This simple type specifies a coordinate within the document. This can be used for measurements or spacing with the maximum size requirement being a 32 bit integer.

The units of measurement used here are EMUs (English Metric 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_Coordinate32Unqualified) is located in §A.4.1. *end note*]

20.1.10.19 ST_CoordinateUnqualified (Coordinate)

This simple type represents a one dimensional position or length in EMUs.

This simple type's contents are a restriction of the W3C XML Schema long datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to -27273042329600.
- This simple type has a maximum value of less than or equal to 27273042316900.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CoordinateUnqualified) is located in §A.4.1. *end note*]

20.1.10.20 ST_DgmBuildStep (Diagram Animation Build Steps)

This simple type specifies an animation build step within a diagram animation.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bg (Background)	Animate the diagram background for this animation build step
sp (Shape)	Animate a diagram shape for this animation build step

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DgmBuildStep) is located in §A.4.1. *end note*]

20.1.10.21 ST_DrawingElementId (Drawing Element ID)

This simple type specifies a unique integer identifier for each drawing element.

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DrawingElementId) is located in §A.4.1. *end note*]

20.1.10.22 ST_EffectContainerType (Effect Container Type)

This simple type determines the relationship between effects in a container, either sibling or tree.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
sib (Sibling)	Each effect is separately applied to the parent object. [*Example*: If the parent element contains an outer shadow and a reflection, the resulting effect is a shadow around the parent object and a reflection of the object. The reflection does not have a shadow. *end example*]
tree (Tree)	Each effect is applied to the result of the previous effect. [*Example*: If the parent element contains an outer shadow followed by a glow, the shadow is first applied to the parent object. Then, the glow is applied to the shadow (rather than the original object). The resulting effect would be a glowing shadow. *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_EffectContainerType) is located in §A.4.1. *end note*]

20.1.10.23 ST_FixedAngle (Fixed Angle)

This simple type represents a fixed range angle in 60000ths of a degree. Range from (-90, 90 degrees).

This simple type's contents are a restriction of the ST_Angle datatype (§20.1.10.3).

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than -5400000.
- This simple type has a maximum value of less than 5400000.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FixedAngle) is located in §A.4.1. *end note*]

20.1.10.24 ST_FixedPercentage (Fixed Percentage)

This simple type represents a fixed percentage from negative one hundred to positive one hundred percent. See the union's member types for details.

This simple type is a union of the following types:

- The ST_FixedPercentage simple type (§22.9.2.3).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FixedPercentage) is located in §A.4.1. *end note*]

20.1.10.25 ST_FontCollectionIndex (Font Collection Index)

This simple type represents one of the fonts associated with the style.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
major (Major Font)	The major font of the style's font scheme.
minor (Minor Font)	The minor font of the style's font scheme.
none (None)	No font reference.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FontCollectionIndex) is located in §A.4.1. *end note*]

20.1.10.26 ST_FOVAngle (Field of View Angle)

Represents a positive angle in 60000ths of a degree. Range from [0, 180] degrees.

This simple type's contents are a restriction of the ST_Angle datatype (§20.1.10.3).

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 10800000.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FOVAngle) is located in §A.4.1. *end note*]

20.1.10.27 ST_GeomGuideFormula (Geometry Guide Formula Properties)

This simple type specifies a geometry guide formula.

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_GeomGuideFormula) is located in §A.4.1. *end note*]

20.1.10.28 ST_GeomGuideName (Geometry Guide Name Properties)

This simple type specifies a geometry guide name.

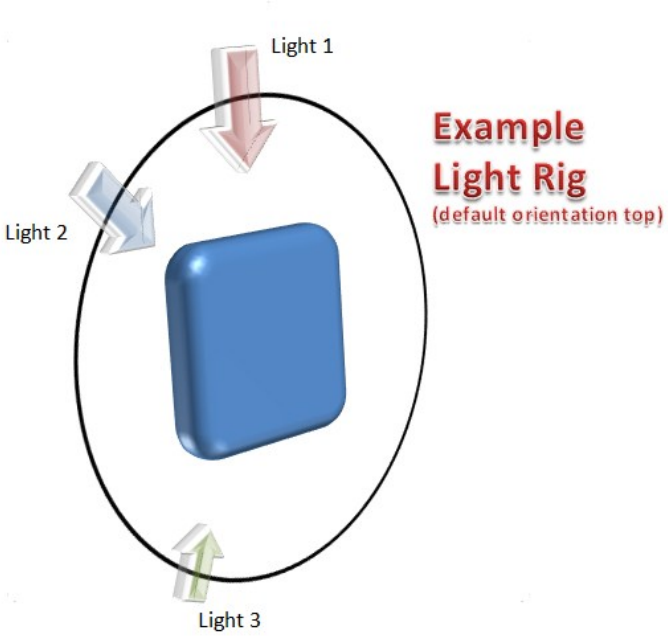
This simple type's contents are a restriction of the W3C XML Schema token datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_GeomGuideName) is located in §A.4.1. *end note*]

20.1.10.29 ST_LightRigDirection (Light Rig Direction)

Represents the direction from which the light rig is positioned relative to the scene. The light rig, itself, can be made up of multiple lights in any orientation around a given shape. This simple type defines the orientation of the light rig as a whole, and not the individual lights within the rig. This means that because the direction of the light rig is left, that does not guarantee the light is coming from the left side of the shape, but rather the orientation of the rig as a whole is rotated to the left.

[*Example*: Consider the following example as a visual representation of a light rig oriented from the top of the shape in the center:



In this example we see that the light rig defines three lights (all in a single plane as represented by the black circular line). The lights defined in this representation can all have different intensities, which means, for this example, Light 3 and Light 2 look to have a more intense effect (or could even be a different color) than Light 1. One can imagine rotating this rig to the so that Light 1 is to the right of the shape when the light rig direction is defined to be right. *end example*]

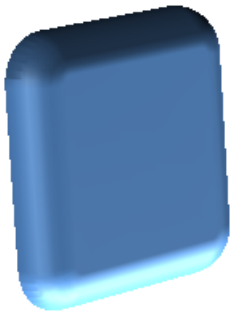
The following properties were used to define the shape used in the image examples below:

- Rounded rectangle shape
- Three Point light rig type
- Circle bevel type
- Plastic material type
- Camera type defined by the `orthographicFront` preset
- Bevel width and height each equal to `190500`

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bottom)	[*Example*: Consider the following example of a light direction from the bottom:  *end example*]
bl (Bottom Left)	[*Example*: Consider the following example of a light direction from the bottom left:  *end example*]
br (Bottom Right)	[*Example*: Consider the following example of a light direction from the bottom right:

Enumeration Value	Description
	 *end example*]
l (Left)	[*Example*: Consider the following example of a light direction from the left:  *end example*]
r (Right)	[*Example*: Consider the following example of a light direction from the right:  *end example*]
t (Top)	[*Example*: Consider the following example of a light

Enumeration Value	Description
	direction from the top:  *end example*]
tl (Top Left)	[*Example*: Consider the following example of a light direction from the top left:  *end example*]
tr (Top Right)	[*Example*: Consider the following example of a light direction from the top right:  *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LightRigDirection) is located in §A.4.1. *end note*]

20.1.10.30 ST_LightRigType (Light Rig Type)

Represents a preset light right that can be applied to a shape. The light rig represents a group of lights oriented in a specific way relative to a 3D scene. The following properties were used to define the shape used in the image examples below:

- Rounded rectangle shape
- Circle bevel type
- Warm Matte material type
- Camera type defined by the `perspectiveContrastingRightFacing` preset
- Bevel width and height each equal to `190500`

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

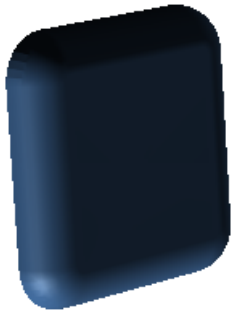
Enumeration Value	Description
balanced (Light Rig Enum (Balanced))	Balanced
brightRoom (Bright Room)	[*Example*: Consider the following example of the `brightRoom` light rig applied to a basic shape:  *end example*]
chilly (Chilly)	[*Example*: Consider the following example of the `chilly` light rig applied to a basic shape:

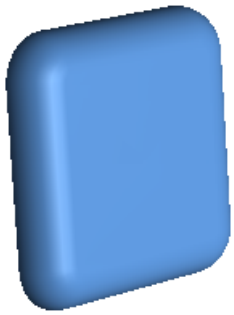
Enumeration Value	Description
	 *end example*]
contrasting (Contrasting)	[*Example*: Consider the following example of the `contrasting` light rig applied to a basic shape:  *end example*]
flat (Flat)	[*Example*: Consider the following example of the `flat` light rig applied to a basic shape:  *end example*]
flood (Flood)	[*Example*: Consider the following example of the `flood` light rig applied to a basic shape:

Enumeration Value	Description
	 *end example*]
freezing (Freezing)	[*Example*: Consider the following example of the `freezing` light rig applied to a basic shape:  *end example*]
glow (Glow)	[*Example*: Consider the following example of the `glow` light rig applied to a basic shape:  *end example*]
harsh (Harsh)	[*Example*: Consider the following example of the

Enumeration Value	Description
	`harsh` light rig applied to a basic shape:  *end example*]
legacyFlat1 (Legacy Flat 1)	[*Example*: Consider the following example of the `legacyFlat1` light rig applied to a basic shape:  *end example*]
legacyFlat2 (Legacy Flat 2)	[*Example*: Consider the following example of the `legacyFlat2` light rig applied to a basic shape:  *end example*]
legacyFlat3 (Legacy Flat 3)	[*Example*: Consider the following example of the

Enumeration Value	Description
	`legacyFlat3` light rig applied to a basic shape:  *end example*]
legacyFlat4 (Legacy Flat 4)	[*Example*: Consider the following example of the `legacyFlat4` light rig applied to a basic shape:  *end example*]
legacyHarsh1 (Legacy Harsh 1)	[*Example*: Consider the following example of the `legacyHarsh1` light rig applied to a basic shape:  *end example*]

Enumeration Value	Description
legacyHarsh2 (Legacy Harsh 2)	[*Example*: Consider the following example of the `legacyHarsh2` light rig applied to a basic shape:  *end example*]
legacyHarsh3 (Legacy Harsh 3)	[*Example*: Consider the following example of the `legacyHarsh3` light rig applied to a basic shape:  *end example*]
legacyHarsh4 (Legacy Harsh 4)	[*Example*: Consider the following example of the `legacyHarsh4` light rig applied to a basic shape:  *end example*]

Enumeration Value	Description
legacyNormal1 (Legacy Normal 1)	[*Example*: Consider the following example of the `legacyNormal1` light rig applied to a basic shape:  *end example*]
legacyNormal2 (Legacy Normal 2)	[*Example*: Consider the following example of the `legacyNormal2` light rig applied to a basic shape:  *end example*]
legacyNormal3 (Legacy Normal 3)	[*Example*: Consider the following example of the `legacyNormal3` light rig applied to a basic shape:  *end example*]

Enumeration Value	Description
legacyNormal4 (Legacy Normal 4)	[*Example*: Consider the following example of the `legacyNormal4` light rig applied to a basic shape:  *end example*]
morning (Morning)	[*Example*: Consider the following example of the `morning` light rig applied to a basic shape:  *end example*]
soft (Soft)	[*Example*: Consider the following example of the `soft` light rig applied to a basic shape:  *end example*]

Enumeration Value	Description
sunrise (Sunrise)	[*Example*: Consider the following example of the `sunrise` light rig applied to a basic shape:  *end example*]
sunset (Sunset)	[*Example*: Consider the following example of the `sunset` light rig applied to a basic shape:  *end example*]
threePt (Three Point)	[*Example*: Consider the following example of the `threePt` light rig applied to a basic shape:  *end example*]

Enumeration Value	Description
twoPt (Two Point)	[*Example*: Consider the following example of the `twoPt` light rig applied to a basic shape:  *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LightRigType) is located in §A.4.1. *end note*]

20.1.10.31 ST_LineCap (End Line Cap)

This simple type specifies how to cap the ends of lines. This also affects the ends of line segments for dashed lines.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
flat (Flat Line Cap)	Line ends at end point.
rnd (Round Line Cap)	Rounded ends. Semi-circle protrudes by half line width.
sq (Square Line Cap)	Square protrudes by half line width.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LineCap) is located in §A.4.1. *end note*]

20.1.10.32 ST_LineEndLength (Line End Length)

This simple type represents the length of the line end decoration (e.g., arrowhead) relative to the width of the line itself.

[*Example:* See the example images below. These samples have an arrow line end type and medium line end width. *end example*]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
lg (Large)	 Large
med (Medium)	 Medium
sm (Small)	 Small

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LineEndLength) is located in §A.4.1. *end note*]

20.1.10.33 ST_LineEndType (Line End Type)

This simple type represents the shape decoration that appears at the ends of lines. For example, one choice is an arrow head.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
arrow (Arrow Head)	 Line arrow head
diamond (Diamond)	 Diamond
none (None)	No end
oval (Oval)	 Oval
stealth (Stealth Arrow)	 Stealth arrow head
triangle (Triangle Arrow Head)	 Triangle arrow head

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LineEndType) is located in §A.4.1. *end note*]

20.1.10.34 ST_LineEndWidth (Line End Width)

This simple type represents the width of the line end decoration (e.g., arrowhead) relative to the width of the line itself.

[*Example:* See the example images below. These samples have an arrow line end type and medium line end length. *end example*]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
lg (Large)	→ Large
med (Medium)	→ Medium
sm (Small)	→ Small

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LineEndWidth) is located in §A.4.1. *end note*]

20.1.10.35 ST_LineWidth (Line Width)

This simple type specifies the width of a line in EMUs. 1 pt = 12700 EMUs.

This simple type's contents are a restriction of the ST_Coordinate32Unqualified datatype (§20.1.10.18).

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 20116800.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LineWidth) is located in §A.4.1. *end note*]

20.1.10.36 ST_OnOffStyleType (On/Off Style Type)

This simple type represents whether a style property should be applied.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
def (Default)	Follow parent settings. For a themed property, follow the theme settings. For an unthemed property, follow the parent setting in the property inheritance chain.
off (Off)	Property is off.
on (On)	Property is on.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_OnOffStyleType) is located in §A.4.1. *end note*]

20.1.10.37 ST_PathFillMode (Path Fill Mode)

This simple type specifies the manner in which a path should be filled. The lightening and darkening of a path allow for certain parts of the shape to be colored lighter of darker depending on user preference.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
darken (Darken Path Fill)	This specifies that the corresponding path should have a darker shaded color applied to it's fill.
darkenLess (Darken Path Fill Less)	This specifies that the corresponding path should have a slightly darker shaded color applied to it's fill.
lighten (Lighten Path Fill)	This specifies that the corresponding path should have a lightly shaded color applied to it's fill.
lightenLess (Lighten Path Fill Less)	This specifies that the corresponding path should have a slightly lighter shaded color applied to it's fill.
none (No Path Fill)	This specifies that the corresponding path should have no fill.
norm (Normal Path Fill)	This specifies that the corresponding path should have a normally shaded color applied to it's fill.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PathFillMode) is located in §A.4.1. *end note*]

20.1.10.38 ST_PathShadeType (Path Shade Type)

This simple type describes the shape of path to follow for a path gradient shade.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
circle (Circle)	Gradient follows a circular path
rect (Rectangle)	Gradient follows a rectangular path
shape (Shape)	Gradient follows the shape

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PathShadeType) is located in §A.4.1. *end note*]

20.1.10.39 ST_PenAlignment (Alignment Type)

This simple type specifies the Pen Alignment type for use within a text body.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ctr (Center Alignment)	Center pen (line drawn at center of path stroke).
in (Inset Alignment)	Inset pen (the pen is aligned on the inside of the edge of the path).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PenAlignment) is located in §A.4.1. *end note*]

20.1.10.40 ST_Percentage (Percentage)

This simple type specifies that its contents will contain a percentage value. See the union's member types for details.

This simple type is a union of the following types:

- The ST_Percentage simple type (§22.9.2.9).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Percentage) is located in §A.4.1. *end note*]

20.1.10.41 ST_PositiveCoordinate (Positive Coordinate)

This simple type represents a positive position or length in EMUs.

This simple type's contents are a restriction of the W3C XML Schema long 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 27273042316900.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PositiveCoordinate) is located in §A.4.1. *end note*]

20.1.10.42 ST_PositiveCoordinate32 (Positive Coordinate Point)

This simple type specifies the a positive coordinate point that has a maximum size of 32 bits.

The units of measurement used here are EMUs (English Metric Units).

This simple type's contents are a restriction of the ST_Coordinate32Unqualified datatype (§20.1.10.18).

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_PositiveCoordinate32) is located in §A.4.1. *end note*]

20.1.10.43 ST_PositiveFixedAngle (Positive Fixed Angle)

This simple type represents a positive angle in 60000ths of a degree. Range from [0, 360 degrees).

This simple type's contents are a restriction of the ST_Angle datatype (§20.1.10.3).

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 21600000.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PositiveFixedAngle) is located in §A.4.1. *end note*]

20.1.10.44 ST_PositiveFixedPercentage (Positive Fixed Percentage)

This simple type specifies that its contents will contain a positive percentage value from zero through one hundred percent. See the union's member types for details.

This simple type is a union of the following types:

- The ST_PositiveFixedPercentage simple type (§22.9.2.10).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PositiveFixedPercentage) is located in §A.4.1. *end note*]

20.1.10.45 ST_PositivePercentage (Positive Percentage Value with Sign)

This simple type specifies that its contents will contain a positive percentage value. See the union's member types for details.

This simple type is a union of the following types:

- The ST_PositivePercentage simple type (§22.9.2.11).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PositivePercentage) is located in §A.4.1. *end note*]

20.1.10.46 ST_PresetCameraType (Preset Camera Type)

These enumeration values represent different algorithmic methods for setting all camera properties, including position. The following example images below are all based off the following shape:

In this image, we can see the shape has a camera pointing directly at the front face.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
isometricBottomDown (Isometric Bottom Down)	[*Example*: Consider the following example of the camera preset type:  *end example*]
isometricBottomUp (Isometric Bottom Up)	[*Example*: Consider the following example of the camera preset type:  *end example*]
isometricLeftDown (Isometric Left Down)	[*Example*: Consider the following example of the camera preset type:

Enumeration Value	Description
	 *end example*]
isometricLeftUp (Isometric Left Up)	[*Example*: Consider the following example of the camera preset type:  *end example*]
isometricOffAxis1Left (Isometric Off Axis 1 Left)	[*Example*: Consider the following example of the camera preset type:  *end example*]

Enumeration Value	Description
isometricOffAxis1Right (Isometric Off Axis 1 Right)	[*Example*: Consider the following example of the camera preset type:  *end example*]
isometricOffAxis1Top (Isometric Off Axis 1 Top)	[*Example*: Consider the following example of the camera preset type:  *end example*]
isometricOffAxis2Left (Isometric Off Axis 2 Left)	[*Example*: Consider the following example of the camera preset type:  *end example*]
isometricOffAxis2Right (Isometric Off Axis 2 Right)	[*Example*: Consider the following example of the camera preset type:

Enumeration Value	Description
	 *end example*]
isometricOffAxis2Top (Isometric Off Axis 2 Top)	[*Example*: Consider the following example of the camera preset type:  *end example*]
isometricOffAxis3Bottom (Isometric Off Axis 3 Bottom)	[*Example*: Consider the following example of the camera preset type:  *end example*]
isometricOffAxis3Left (Isometric Off Axis 3 Left)	[*Example*: Consider the following example of the camera preset type:

Enumeration Value	Description
	 *end example*]
isometricOffAxis3Right (Isometric Off Axis 3 Right)	[*Example*: Consider the following example of the camera preset type:  *end example*]
isometricOffAxis4Bottom (Isometric Off Axis 4 Bottom)	[*Example*: Consider the following example of the camera preset type:  *end example*]
isometricOffAxis4Left (Isometric Off Axis 4 Left)	[*Example*: Consider the following example of the camera preset type:

Enumeration Value	Description
	 *end example*]
isometricOffAxis4Right (Isometric Off Axis 4 Right)	[*Example*: Consider the following example of the camera preset type:  *end example*]
isometricRightDown (Isometric Right Down)	[*Example*: Consider the following example of the camera preset type:  *end example*]

Enumeration Value	Description
isometricRightUp (Isometric Right Up)	[*Example*: Consider the following example of the camera preset type:  *end example*]
isometricTopDown (Isometric Top Down)	[*Example*: Consider the following example of the camera preset type:  *end example*]
isometricTopUp (Isometric Top Up)	[*Example*: Consider the following example of the camera preset type:  *end example*]
legacyObliqueBottom (Legacy Oblique Bottom)	[*Example*: Consider the following example of the

Enumeration Value	Description
	camera preset type:  *end example*]
legacyObliqueBottomLeft (Legacy Oblique Bottom Left)	[*Example*: Consider the following example of the camera preset type:  *end example*]
legacyObliqueBottomRight (Legacy Oblique Bottom Right)	[*Example*: Consider the following example of the camera preset type:  *end example*]
legacyObliqueFront (Legacy Oblique Front)	[*Example*: Consider the following example of the camera preset type:

Enumeration Value	Description
	 *end example*]
legacyObliqueLeft (Legacy Oblique Left)	[*Example*: Consider the following example of the camera preset type:  *end example*]
legacyObliqueRight (Legacy Oblique Right)	[*Example*: Consider the following example of the camera preset type:  *end example*]
legacyObliqueTop (Legacy Oblique Top)	[*Example*: Consider the following example of the camera preset type:

Enumeration Value	Description
	 *end example*]
legacyObliqueTopLeft (Legacy Oblique Top Left)	[*Example*: Consider the following example of the camera preset type:  *end example*]
legacyObliqueTopRight (Legacy Oblique Top Right)	[*Example*: Consider the following example of the camera preset type:  *end example*]
legacyPerspectiveBottom (Legacy Perspective Bottom)	[*Example*: Consider the following example of the camera preset type:

Enumeration Value	Description
	 *end example*]
legacyPerspectiveBottomLeft (Legacy Perspective Bottom Left)	[*Example*: Consider the following example of the camera preset type:  *end example*]
legacyPerspectiveBottomRight (Legacy Perspective Bottom Right)	[*Example*: Consider the following example of the camera preset type:  *end example*]
legacyPerspectiveFront (Legacy Perspective Front)	[*Example*: Consider the following example of the camera preset type:

Enumeration Value	Description
	 *end example*]
legacyPerspectiveLeft (Legacy Perspective Left)	[*Example*: Consider the following example of the camera preset type:  *end example*]
legacyPerspectiveRight (Legacy Perspective Right)	[*Example*: Consider the following example of the camera preset type:  *end example*]
legacyPerspectiveTop (Legacy Perspective Top)	[*Example*: Consider the following example of the camera preset type:

Enumeration Value	Description
	 *end example*]
legacyPerspectiveTopLeft (Legacy Perspective Top Left)	[*Example*: Consider the following example of the camera preset type:  *end example*]
legacyPerspectiveTopRight (Legacy Perspective Top Right)	[*Example*: Consider the following example of the camera preset type:  *end example*]
obliqueBottom (Oblique Bottom)	[*Example*: Consider the following example of the camera preset type:

Enumeration Value	Description
	 *end example*]
obliqueBottomLeft (Oblique Bottom Left)	[*Example*: Consider the following example of the camera preset type:  *end example*]
obliqueBottomRight (Oblique Bottom Right)	[*Example*: Consider the following example of the camera preset type:  *end example*]
obliqueLeft (Oblique Left)	[*Example*: Consider the following example of the camera preset type:

Enumeration Value	Description
	 *end example*]
obliqueRight (Oblique Right)	[*Example*: Consider the following example of the camera preset type:  *end example*]
obliqueTop (Oblique Top)	[*Example*: Consider the following example of the camera preset type:  *end example*]
obliqueTopLeft (Oblique Top Left)	[*Example*: Consider the following example of the camera preset type:

Enumeration Value	Description
	 *end example*]
obliqueTopRight (Oblique Top Right)	[*Example*: Consider the following example of the camera preset type:  *end example*]
orthographicFront (Orthographic Front)	[*Example*: Consider the following example of the camera preset type:  *end example*]
perspectiveAbove (Orthographic Above)	[*Example*: Consider the following example of the camera preset type:

Enumeration Value	Description
	 *end example*]
perspectiveAboveLeftFacing (Perspective Above Left Facing)	[*Example*: Consider the following example of the camera preset type:  *end example*]
perspectiveAboveRightFacing (Perspective Above Right Facing)	[*Example*: Consider the following example of the camera preset type:  *end example*]
perspectiveBelow (Perspective Below)	[*Example*: Consider the following example of the

Enumeration Value	Description
	camera preset type:  *end example*]
perspectiveContrastingLeftFacing (Perspective Contrasting Left Facing)	[*Example*: Consider the following example of the camera preset type:  *end example*]
perspectiveContrastingRightFacing (Perspective Contrasting Right Facing)	[*Example*: Consider the following example of the camera preset type:  *end example*]
perspectiveFront (Perspective Front)	[*Example*: Consider the following example of the

Enumeration Value	Description
	camera preset type:  *end example*]
perspectiveHeroicExtremeLeftFacing (Perspective Heroic Extreme Left Facing)	[*Example*: Consider the following example of the camera preset type:  *end example*]
perspectiveHeroicExtremeRightFacing (Perspective Heroic Extreme Right Facing)	[*Example*: Consider the following example of the camera preset type:  *end example*]
perspectiveHeroicLeftFacing (Perspective Heroic Left Facing)	[*Example*: Consider the following example of the camera preset type:

Enumeration Value	Description
	 *end example*]
perspectiveHeroicRightFacing (Perspective Heroic Right Facing)	[*Example*: Consider the following example of the camera preset type:  *end example*]
perspectiveLeft (Perspective Left)	[*Example*: Consider the following example of the camera preset type:  *end example*]
perspectiveRelaxed (Perspective Relaxed)	[*Example*: Consider the following example of the camera preset type:

Enumeration Value	Description
	 *end example*]
perspectiveRelaxedModerately (Perspective Relaxed Moderately)	[*Example*: Consider the following example of the camera preset type:  *end example*]
perspectiveRight (Perspective Right)	[*Example*: Consider the following example of the camera preset type:  *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PresetCameraType) is located in §A.4.1. *end note*]

20.1.10.47 ST_PresetColorVal (Preset Color Value)

This simple type represents a preset color value.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
aliceBlue (Alice Blue Preset Color)	Specifies a color with RGB value (240,248,255)
antiqueWhite (Antique White Preset Color)	Specifies a color with RGB value (250,235,215)
aqua (Aqua Preset Color)	Specifies a color with RGB value (0,255,255)
aquamarine (Aquamarine Preset Color)	Specifies a color with RGB value (127,255,212)
azure (Azure Preset Color)	Specifies a color with RGB value (240,255,255)
beige (Beige Preset Color)	Specifies a color with RGB value (245,245,220)
bisque (Bisque Preset Color)	Specifies a color with RGB value (255,228,196)
black (Black Preset Color)	Specifies a color with RGB value (0,0,0)
blanchedAlmond (Blanched Almond Preset Color)	Specifies a color with RGB value (255,235,205)
blue (Blue Preset Color)	Specifies a color with RGB value (0,0,255)
blueViolet (Blue Violet Preset Color)	Specifies a color with RGB value (138,43,226)
brown (Brown Preset Color)	Specifies a color with RGB value (165,42,42)
burlyWood (Burly Wood Preset Color)	Specifies a color with RGB value (222,184,135)
cadetBlue (Cadet Blue Preset Color)	Specifies a color with RGB value (95,158,160)
chartreuse (Chartreuse Preset Color)	Specifies a color with RGB value (127,255,0)
chocolate (Chocolate Preset Color)	Specifies a color with RGB value (210,105,30)
coral (Coral Preset Color)	Specifies a color with RGB value (255,127,80)
cornflowerBlue (Cornflower Blue Preset Color)	Specifies a color with RGB value (100,149,237)
cornsilk (Cornsilk Preset Color)	Specifies a color with RGB value (255,248,220)
crimson (Crimson Preset Color)	Specifies a color with RGB value (220,20,60)
cyan (Cyan Preset Color)	Specifies a color with RGB value (0,255,255)
darkBlue (Dark Blue Preset Color)	Specifies a color with RGB value (0,0,139)
darkCyan (Dark Cyan Preset Color)	Specifies a color with RGB value (0,139,139)
darkGoldenrod (Dark Goldenrod Preset Color)	Specifies a color with RGB value (184,134,11)
darkGray (Dark Gray Preset Color)	Specifies a color with RGB value (169,169,169)
darkGreen (Dark Green Preset Color)	Specifies a color with RGB value (0,100,0)
darkGrey (Dark Gray Preset Color)	Specifies a color with RGB value (169,169,169)
darkKhaki (Dark Khaki Preset Color)	Specifies a color with RGB value (189,183,107)
darkMagenta (Dark Magenta Preset Color)	Specifies a color with RGB value (139,0,139)
darkOliveGreen (Dark Olive Green Preset Color)	Specifies a color with RGB value (85,107,47)
darkOrange (Dark Orange Preset Color)	Specifies a color with RGB value (255,140,0)
darkOrchid (Dark Orchid Preset Color)	Specifies a color with RGB value (153,50,204)

Enumeration Value	Description
darkRed (Dark Red Preset Color)	Specifies a color with RGB value (139,0,0)
darkSalmon (Dark Salmon Preset Color)	Specifies a color with RGB value (233,150,122)
darkSeaGreen (Dark Sea Green Preset Color)	Specifies a color with RGB value (143,188,143)
darkSlateBlue (Dark Slate Blue Preset Color)	Specifies a color with RGB value (72,61,139)
darkSlateGray (Dark Slate Gray Preset Color)	Specifies a color with RGB value (47,79,79)
darkSlateGrey (Dark Slate Gray Preset Color)	Specifies a color with RGB value (47,79,79)
darkTurquoise (Dark Turquoise Preset Color)	Specifies a color with RGB value (0,206,209)
darkViolet (Dark Violet Preset Color)	Specifies a color with RGB value (148,0,211)
deepPink (Deep Pink Preset Color)	Specifies a color with RGB value (255,20,147)
deepSkyBlue (Deep Sky Blue Preset Color)	Specifies a color with RGB value (0,191,255)
dimGray (Dim Gray Preset Color)	Specifies a color with RGB value (105,105,105)
dimGrey (Dim Gray Preset Color)	Specifies a color with RGB value (105,105,105)
dkBlue (Dark Blue Preset Color)	Specifies a color with RGB value (0,0,139)
dkCyan (Dark Cyan Preset Color)	Specifies a color with RGB value (0,139,139)
dkGoldenrod (Dark Goldenrod Preset Color)	Specifies a color with RGB value (184,134,11)
dkGray (Dark Gray Preset Color)	Specifies a color with RGB value (169,169,169)
dkGreen (Dark Green Preset Color)	Specifies a color with RGB value (0,100,0)
dkGrey (Dark Gray Preset Color)	Specifies a color with RGB value (169,169,169)
dkKhaki (Dark Khaki Preset Color)	Specifies a color with RGB value (189,183,107)
dkMagenta (Dark Magenta Preset Color)	Specifies a color with RGB value (139,0,139)
dkOliveGreen (Dark Olive Green Preset Color)	Specifies a color with RGB value (85,107,47)
dkOrange (Dark Orange Preset Color)	Specifies a color with RGB value (255,140,0)
dkOrchid (Dark Orchid Preset Color)	Specifies a color with RGB value (153,50,204)
dkRed (Dark Red Preset Color)	Specifies a color with RGB value (139,0,0)
dkSalmon (Dark Salmon Preset Color)	Specifies a color with RGB value (233,150,122)
dkSeaGreen (Dark Sea Green Preset Color)	Specifies a color with RGB value (143,188,139)
dkSlateBlue (Dark Slate Blue Preset Color)	Specifies a color with RGB value (72,61,139)
dkSlateGray (Dark Slate Gray Preset Color)	Specifies a color with RGB value (47,79,79)
dkSlateGrey (Dark Slate Gray Preset Color)	Specifies a color with RGB value (47,79,79)
dkTurquoise (Dark Turquoise Preset Color)	Specifies a color with RGB value (0,206,209)
dkViolet (Dark Violet Preset Color)	Specifies a color with RGB value (148,0,211)
dodgerBlue (Dodger Blue Preset Color)	Specifies a color with RGB value (30,144,255)
firebrick (Firebrick Preset Color)	Specifies a color with RGB value (178,34,34)
floralWhite (Floral White Preset Color)	Specifies a color with RGB value (255,250,240)

Enumeration Value	Description
forestGreen (Forest Green Preset Color)	Specifies a color with RGB value (34,139,34)
fuchsia (Fuchsia Preset Color)	Specifies a color with RGB value (255,0,255)
gainsboro (Gainsboro Preset Color)	Specifies a color with RGB value (220,220,220)
ghostWhite (Ghost White Preset Color)	Specifies a color with RGB value (248,248,255)
gold (Gold Preset Color)	Specifies a color with RGB value (255,215,0)
goldenrod (Goldenrod Preset Color)	Specifies a color with RGB value (218,165,32)
gray (Gray Preset Color)	Specifies a color with RGB value (128,128,128)
green (Green Preset Color)	Specifies a color with RGB value (0,128,0)
greenYellow (Green Yellow Preset Color)	Specifies a color with RGB value (173,255,47)
grey (Gray Preset Color)	Specifies a color with RGB value (128,128,128)
honeydew (Honeydew Preset Color)	Specifies a color with RGB value (240,255,240)
hotPink (Hot Pink Preset Color)	Specifies a color with RGB value (255,105,180)
indianRed (Indian Red Preset Color)	Specifies a color with RGB value (205,92,92)
indigo (Indigo Preset Color)	Specifies a color with RGB value (75,0,130)
ivory (Ivory Preset Color)	Specifies a color with RGB value (255,255,240)
khaki (Khaki Preset Color)	Specifies a color with RGB value (240,230,140)
lavender (Lavender Preset Color)	Specifies a color with RGB value (230,230,250)
lavenderBlush (Lavender Blush Preset Color)	Specifies a color with RGB value (255,240,245)
lawnGreen (Lawn Green Preset Color)	Specifies a color with RGB value (124,252,0)
lemonChiffon (Lemon Chiffon Preset Color)	Specifies a color with RGB value (255,250,205)
lightBlue (Light Blue Preset Color)	Specifies a color with RGB value (173,216,230)
lightCoral (Light Coral Preset Color)	Specifies a color with RGB value (240,128,128)
lightCyan (Light Cyan Preset Color)	Specifies a color with RGB value (224,255,255)
lightGoldenrodYellow (Light Goldenrod Yellow Preset Color)	Specifies a color with RGB value (250,250,210)
lightGray (Light Gray Preset Color)	Specifies a color with RGB value (211,211,211)
lightGreen (Light Green Preset Color)	Specifies a color with RGB value (144,238,144)
lightGrey (Light Gray Preset Color)	Specifies a color with RGB value (211,211,211)
lightPink (Light Pink Preset Color)	Specifies a color with RGB value (255,182,193)
lightSalmon (Light Salmon Preset Color)	Specifies a color with RGB value (255,160,122)
lightSeaGreen (Light Sea Green Preset Color)	Specifies a color with RGB value (32,178,170)
lightSkyBlue (Light Sky Blue Preset Color)	Specifies a color with RGB value (135,206,250)
lightSlateGray (Light Slate Gray Preset Color)	Specifies a color with RGB value (119,136,153)
lightSlateGrey (Light Slate Gray Preset Color)	Specifies a color with RGB value (119,136,153)

Enumeration Value	Description
lightSteelBlue (Light Steel Blue Preset Color)	Specifies a color with RGB value (176,196,222)
lightYellow (Light Yellow Preset Color)	Specifies a color with RGB value (255,255,224)
lime (Lime Preset Color)	Specifies a color with RGB value (0,255,0)
limeGreen (Lime Green Preset Color)	Specifies a color with RGB value (50,205,50)
linen (Linen Preset Color)	Specifies a color with RGB value (250,240,230)
ltBlue (Light Blue Preset Color)	Specifies a color with RGB value (173,216,230)
ltCoral (Light Coral Preset Color)	Specifies a color with RGB value (240,128,128)
ltCyan (Light Cyan Preset Color)	Specifies a color with RGB value (224,255,255)
ltGoldenrodYellow (Light Goldenrod Yellow Preset Color)	Specifies a color with RGB value (250,250,120)
ltGray (Light Gray Preset Color)	Specifies a color with RGB value (211,211,211)
ltGreen (Light Green Preset Color)	Specifies a color with RGB value (144,238,144)
ltGrey (Light Gray Preset Color)	Specifies a color with RGB value (211,211,211)
ltPink (Light Pink Preset Color)	Specifies a color with RGB value (255,182,193)
ltSalmon (Light Salmon Preset Color)	Specifies a color with RGB value (255,160,122)
ltSeaGreen (Light Sea Green Preset Color)	Specifies a color with RGB value (32,178,170)
ltSkyBlue (Light Sky Blue Preset Color)	Specifies a color with RGB value (135,206,250)
ltSlateGray (Light Slate Gray Preset Color)	Specifies a color with RGB value (119,136,153)
ltSlateGrey (Light Slate Gray Preset Color)	Specifies a color with RGB value (119,136,153)
ltSteelBlue (Light Steel Blue Preset Color)	Specifies a color with RGB value (176,196,222)
ltYellow (Light Yellow Preset Color)	Specifies a color with RGB value (255,255,224)
magenta (Magenta Preset Color)	Specifies a color with RGB value (255,0,255)
maroon (Maroon Preset Color)	Specifies a color with RGB value (128,0,0)
medAquamarine (Medium Aquamarine Preset Color)	Specifies a color with RGB value (102,205,170)
medBlue (Medium Blue Preset Color)	Specifies a color with RGB value (0,0,205)
mediumAquamarine (Medium Aquamarine Preset Color)	Specifies a color with RGB value (102,205,170)
mediumBlue (Medium Blue Preset Color)	Specifies a color with RGB value (0,0,205)
mediumOrchid (Medium Orchid Preset Color)	Specifies a color with RGB value (186,85,211)
mediumPurple (Medium Purple Preset Color)	Specifies a color with RGB value (147,112,219)
mediumSeaGreen (Medium Sea Green Preset Color)	Specifies a color with RGB value (60,179,113)
mediumSlateBlue (Medium Slate Blue Preset Color)	Specifies a color with RGB value (123,104,238)
mediumSpringGreen (Medium Spring Green Preset Color)	Specifies a color with RGB value (0,250,154)
mediumTurquoise (Medium Turquoise Preset Color)	Specifies a color with RGB value (72,209,204)

Enumeration Value	Description
mediumVioletRed (Medium Violet Red Preset Color)	Specifies a color with RGB value (199,21,133)
medOrchid (Medium Orchid Preset Color)	Specifies a color with RGB value (186,85,211)
medPurple (Medium Purple Preset Color)	Specifies a color with RGB value (147,112,219)
medSeaGreen (Medium Sea Green Preset Color)	Specifies a color with RGB value (60,179,113)
medSlateBlue (Medium Slate Blue Preset Color)	Specifies a color with RGB value (123,104,238)
medSpringGreen (Medium Spring Green Preset Color)	Specifies a color with RGB value (0,250,154)
medTurquoise (Medium Turquoise Preset Color)	Specifies a color with RGB value (72,209,204)
medVioletRed (Medium Violet Red Preset Color)	Specifies a color with RGB value (199,21,133)
midnightBlue (Midnight Blue Preset Color)	Specifies a color with RGB value (25,25,112)
mintCream (Mint Cream Preset Color)	Specifies a color with RGB value (245,255,250)
mistyRose (Misty Rose Preset Color)	Specifies a color with RGB value (255,228,225)
moccasin (Moccasin Preset Color)	Specifies a color with RGB value (255,228,181)
navajoWhite (Navajo White Preset Color)	Specifies a color with RGB value (255,222,173)
navy (Navy Preset Color)	Specifies a color with RGB value (0,0,128)
oldLace (Old Lace Preset Color)	Specifies a color with RGB value (253,245,230)
olive (Olive Preset Color)	Specifies a color with RGB value (128,128,0)
oliveDrab (Olive Drab Preset Color)	Specifies a color with RGB value (107,142,35)
orange (Orange Preset Color)	Specifies a color with RGB value (255,165,0)
orangeRed (Orange Red Preset Color)	Specifies a color with RGB value (255,69,0)
orchid (Orchid Preset Color)	Specifies a color with RGB value (218,112,214)
paleGoldenrod (Pale Goldenrod Preset Color)	Specifies a color with RGB value (238,232,170)
paleGreen (Pale Green Preset Color)	Specifies a color with RGB value (152,251,152)
paleTurquoise (Pale Turquoise Preset Color)	Specifies a color with RGB value (175,238,238)
paleVioletRed (Pale Violet Red Preset Color)	Specifies a color with RGB value (219,112,147)
papayaWhip (Papaya Whip Preset Color)	Specifies a color with RGB value (255,239,213)
peachPuff (Peach Puff Preset Color)	Specifies a color with RGB value (255,218,185)
peru (Peru Preset Color)	Specifies a color with RGB value (205,133,63)
pink (Pink Preset Color)	Specifies a color with RGB value (255,192,203)
plum (Plum Preset Color)	Specifies a color with RGB value (221,160,221)
powderBlue (Powder Blue Preset Color)	Specifies a color with RGB value (176,224,230)
purple (Purple Preset Color)	Specifies a color with RGB value (128,0,128)
red (Red Preset Color)	Specifies a color with RGB value (255,0,0)
rosyBrown (Rosy Brown Preset Color)	Specifies a color with RGB value (188,143,143)
royalBlue (Royal Blue Preset Color)	Specifies a color with RGB value (65,105,225)

Enumeration Value	Description
saddleBrown (Saddle Brown Preset Color)	Specifies a color with RGB value (139,69,19)
salmon (Salmon Preset Color)	Specifies a color with RGB value (250,128,114)
sandyBrown (Sandy Brown Preset Color)	Specifies a color with RGB value (244,164,96)
seaGreen (Sea Green Preset Color)	Specifies a color with RGB value (46,139,87)
seaShell (Sea Shell Preset Color)	Specifies a color with RGB value (255,245,238)
sienna (Sienna Preset Color)	Specifies a color with RGB value (160,82,45)
silver (Silver Preset Color)	Specifies a color with RGB value (192,192,192)
skyBlue (Sky Blue Preset Color)	Specifies a color with RGB value (135,206,235)
slateBlue (Slate Blue Preset Color)	Specifies a color with RGB value (106,90,205)
slateGray (Slate Gray Preset Color)	Specifies a color with RGB value (112,128,144)
slateGrey (Slate Gray Preset Color)	Specifies a color with RGB value (112,128,144)
snow (Snow Preset Color)	Specifies a color with RGB value (255,250,250)
springGreen (Spring Green Preset Color)	Specifies a color with RGB value (0,255,127)
steelBlue (Steel Blue Preset Color)	Specifies a color with RGB value (70,130,180)
tan (Tan Preset Color)	Specifies a color with RGB value (210,180,140)
teal (Teal Preset Color)	Specifies a color with RGB value (0,128,128)
thistle (Thistle Preset Color)	Specifies a color with RGB value (216,191,216)
tomato (Tomato Preset Color)	Specifies a color with RGB value (255,99,71)
turquoise (Turquoise Preset Color)	Specifies a color with RGB value (64,224,208)
violet (Violet Preset Color)	Specifies a color with RGB value (238,130,238)
wheat (Wheat Preset Color)	Specifies a color with RGB value (245,222,179)
white (White Preset Color)	Specifies a color with RGB value (255,255,255)
whiteSmoke (White Smoke Preset Color)	Specifies a color with RGB value (245,245,245)
yellow (Yellow Preset Color)	Specifies a color with RGB value (255,255,0)
yellowGreen (Yellow Green Preset Color)	Specifies a color with RGB value (154,205,50)

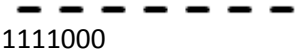
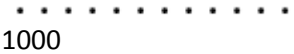
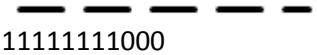
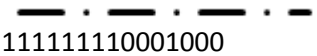
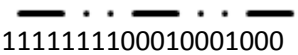
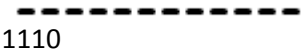
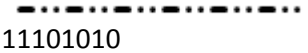
[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PresetColorVal) is located in §A.4.1. *end note*]

20.1.10.48 ST_PresetLineDashVal (Preset Line Dash Value)

This simple type represents preset line dash values. The description for each style shows an illustration of the line style. Each style also contains a precise binary representation of the repeating dash style. Each 1 corresponds to a line segment of the same length as the line width, and each 0 corresponds to a space of the same length as the line width.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
dash (Dash)	 1111000
dashDot (Dash Dot)	11110001000
dot (Dot)	 1000
lgDash (Large Dash)	 11111111000
lgDashDot (Large Dash Dot)	 111111110001000
lgDashDotDot (Large Dash Dot Dot)	 1111111100010001000
solid (Solid)	1
sysDash (System Dash)	 1110
sysDashDot (System Dash Dot)	 111010
sysDashDotDot (System Dash Dot Dot)	 11101010
sysDot (System Dot)	 10

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PresetLineDashVal) is located in §A.4.1. *end note*]

20.1.10.49 ST_PresetMaterialType (Preset Material Type)

Describes surface appearance of a shape. The material type combines with lighting characteristics to create the final look and feel of a shape. The set of material properties which can be combined together to create the presets below consist of the following characteristics:

- Specular color – This defines the color of the highlight associated with the material.
- Specular power – This defines the size and how intense the highlight is. Smaller values provide a larger, but less intense highlight, while larger values provide a smaller, but more intense highlight.
- Diffuse color – This defines the perceived color of the material where an object is directly illuminated by a light source. Generally speaking, the default color here would be based on the shape fill color.

- Ambient color – This defines the perceived color of the material where an object is not directly illuminated by a light source. Generally speaking, the default color here would be based on the shape fill color.
- Emissive color – This defines the color of a light which is perceived to be given off by an object.
- Diffuse Fresnel effect – This is an effect that either darkens (approaches black) or lightens (approaches white) the diffuse color of the material type at glancing angles. Positive values cause the material to become brighter, negative values cause the material to become darker.
- Alpha Fresnel effect – This is an effect that either makes the material more opaque or more transparent at glancing angles. Positive values cause the material to become more opaque, negative values cause the material to become more transparent.

In the following examples, the exact values given for certain properties should be understood to be relative values in order to provide a reference. These values could be different depending upon technologies used to render the material types. The following properties were used to define the shape used in the image examples below:

- Rounded rectangle shape
- Circle bevel type
- Three Point light rig type
- Camera type defined by the `perspectiveContrastingRightFacing` preset
- Bevel width and height each equal to `190500`

This simple type's contents are a restriction of the W3C XML Schema token datatype.

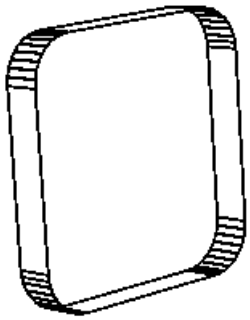
This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
clear (Clear)	The `clear` material type has the following characteristics: • Specular Color: light gray • Diffuse Color: shape fill color with 90% alpha • Ambient Color: shape fill color with 90% alpha • Emissive Color: black • Diffuse Fresnel value: -8 • Alpha Fresnel value: 1 [*Example*: Consider the following example of the `clear` material type:

Enumeration Value	Description
	 *end example*]
dkEdge (Dark Edge)	The dkEdge material type has the following characteristics: • Specular Color: white • Specular Power value: 35 • Ambient Color: shape fill color • Emissive Color: black • Diffuse Fresnel value: -2 [*Example*: Consider the following example of the dkEdge material type:  *end example*]
flat (Flat)	The flat material type has the following characteristics: • Specular Color: very light gray • Specular Power value: 50 • Diffuse Color: black • Ambient Color: black • Emissive Color: shape fill color • Diffuse Fresnel value: -4

Enumeration Value	Description
	[*Example*: Consider the following example of the `flat` material type:  *end example*]
legacyMatte (Legacy Matte)	The `legacyMatte` material type has the following characteristics: • Specular Color: black • Ambient Color: shape fill color • Emissive Color: black • Diffuse Fresnel value: -4 [*Example*: Consider the following example of the `legacyMatte` material type:  *end example*]
legacyMetal (Legacy Metal)	The `legacyMetal` material type has the following characteristics: • Specular Color: shape fill color • Specular Power value: 32 • Diffuse Color: shape fill color darkened by adding black • Ambient Color: shape fill color darkened by adding black

Enumeration Value	Description
	• Emissive Color: black [*Example*: Consider the following example of the `legacyMetal` material type:  *end example*]
legacyPlastic (Legacy Plastic)	The `legacyPlastic` material type has the following characteristics: • Specular Color: white • Specular Power value: 32 • Ambient Color: shape fill color • Emissive Color: black [*Example*: Consider the following example of the `legacyPlastic` material type:  *end example*]
legacyWireframe (Legacy Wireframe)	The `legacyWireframe` material type has none of the associated material properties and is based on a wireframe interpretation of the shape. [*Example*: Consider the following example of the `legacyWireframe` material type:

Enumeration Value	Description
	 *end example*]
matte (Matte)	The `matte` material type has the following characteristics: • Specular Color: black • Ambient Color: shape fill color • Emissive Color: black [*Example*: Consider the following example of the `matte` material type:  *end example*]
metal (Metal)	The `metal` material type has the following characteristics: • Specular Color: shape fill color plus white, which is brightened by 1.5 times the normal value • Specular Power value: 12 • Ambient Color: shape fill color • Emissive Color: black • Diffuse Fresnel value: 4 [*Example*: Consider the following example of the

Enumeration Value	Description
	`metal` material type:  *end example*]
plastic (Plastic)	The `plastic` material type has the following characteristics: • Specular Color: light gray • Specular Power value: 12 • Ambient Color: shape fill color • Emissive Color: black [*Example*: Consider the following example of the `plastic` material type:  *end example*]
powder (Powder)	The `powder` material type has the following characteristics: • Specular Color: dark gray • Specular Power value: 10 • Diffuse Color: gray • Ambient Color: gray • Emissive Color: black • Diffuse Fresnel value: 2

Enumeration Value	Description
	[*Example*: Consider the following example of the `powder` material type:  *end example*]
softEdge (Soft Edge)	The `softEdge` material type has the following characteristics: • Specular Color: white • Specular Power value: 35 • Ambient Color: shape fill color • Emissive Color: black • Diffuse Fresnel value: 4 • Alpha Fresnel value: -10 [*Example*: Consider the following example of the `softEdge` material type:  *end example*]
softmetal (Soft Metal)	The `softMetal` material type has the following characteristics: • Specular Color: shape fill color lightened with white by 50% • Specular Power value: 8

Enumeration Value	Description
	• Ambient Color: shape fill color • Emissive Color: black [*Example*: Consider the following example of the `softmetal` material type:  *end example*]
translucentPowder (Translucent Powder)	The `translucentPowder` material type has the following characteristics: • Specular Color: dark gray • Specular Power value: 10 • Diffuse Color: shape fill color with 30% transparency • Ambient Color: shape fill color with 30% transparency • Emissive Color: black • Diffuse Fresnel value: 2 • Alpha Fresnel value: -1 [*Example*: Consider the following example of the `translucentPowder` material type:  *end example*]

Enumeration Value	Description
warmMatte (Warm Matte)	The `warmMatte` material type has the following characteristics: • Specular Color: dark gray • Specular Power value: 8 • Ambient Color: shape fill color • Emissive Color: black [*Example*: Consider the following example of the `warmMatte` material type:  *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PresetMaterialType) is located in §A.4.1. *end note*]

20.1.10.50 ST_PresetPatternVal (Preset Pattern Value)

This simple type indicates a preset type of pattern fill. The description of each value contains an illustration of the fill type.

[*Note*: These presets correspond to members of the HatchStyle enumeration in the Microsoft .NET Framework. A reference for this type can be found at http://msdn2.microsoft.com/en-us/library/system.drawing.drawing2d.hatchstyle.aspx. *end note*]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
cross (Cross)	
dashDnDiag (Dashed Downward Diagonal)	

Enumeration Value	Description
dashHorz (Dashed Horizontal)	
dashUpDiag (Dashed Upward DIagonal)	
dashVert (Dashed Vertical)	
diagBrick (Diagonal Brick)	
diagCross (Diagonal Cross)	
divot (Divot)	
dkDnDiag (Dark Downward Diagonal)	
dkHorz (Dark Horizontal)	
dkUpDiag (Dark Upward Diagonal)	
dkVert (Dark Vertical)	
dnDiag (Downward Diagonal)	
dotDmnd (Dotted Diamond)	
dotGrid (Dotted Grid)	
horz (Horizontal)	
horzBrick (Horizontal Brick)	
lgCheck (Large Checker Board)	
lgConfetti (Large Confetti)	
lgGrid (Large Grid)	
ltDnDiag (Light Downward Diagonal)	
ltHorz (Light Horizontal)	

Enumeration Value	Description
ltUpDiag (Light Upward Diagonal)	
ltVert (Light Vertical)	
narHorz (Narrow Horizontal)	
narVert (Narrow Vertical)	
openDmnd (Open Diamond)	
pct10 (10%)	
pct20 (20%)	
pct25 (25%)	
pct30 (30%)	
pct40 (40%)	
pct5 (5%)	
pct50 (50%)	
pct60 (60%)	
pct70 (70%)	
pct75 (75%)	
pct80 (80%)	
pct90 (90%)	
plaid (Plaid)	
shingle (Shingle)	
smCheck (Small Checker Board)	

Enumeration Value	Description
smConfetti (Small Confetti)	
smGrid (Small Grid)	
solidDmnd (Solid Diamond)	
sphere (Sphere)	
trellis (Trellis)	
upDiag (Upward Diagonal)	
vert (Vertical)	
wave (Wave)	
wdDnDiag (Wide Downward Diagonal)	
wdUpDiag (Wide Upward Diagonal)	
weave (Weave)	
zigZag (Zig Zag)	

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PresetPatternVal) is located in §A.4.1. *end note*]

20.1.10.51 ST_PresetShadowVal (Preset Shadow Type)

This simple type indicates one of 20 preset shadow types. Each enumeration value description illustrates the type of shadow represented by the value. Each description contains the parameters to the outer shadow effect represented by the preset, in addition to those attributes common to all prstShdw effects.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
shdw1 (Top Left Drop Shadow)	 No additional attributes specified.

Enumeration Value	Description
shdw10 (Top Left Large Drop Shadow)	 align = "br" sx = 125% sy = 125%
shdw11 (Back Left Long Perspective Shadow)	 align = "b" kx = 40.89° sy = 50%
shdw12 (Back Right Long Perspective Shadow)	 align = "b" kx = -40.89° sy = 50%
shdw13 (Top Left Double Drop Shadow)	 Equivalent to two outer shadow effects. Shadow 1: No additional attributes specified. Shadow 2: color = min(1, shadow 1's color (0 <= r, g, b <= 1) + 102/255), per r, g, b component dist = 2 * shadow 1's distance
shdw14 (Bottom Right Small Drop Shadow)	 No additional attributes specified.
shdw15 (Front Left Long Perspective Shadow)	 align = "b" kx = 40.89° sy = -50%
shdw16 (Front Right LongPerspective Shadow)	 align = "b" kx = -40.89° sy = -50%
shdw17 (3D Outer Box Shadow)	 Equivalent to two outer shadow effects. Shadow 1: No additional attributes specified.

Enumeration Value	Description
	Shadow 2: color = min(1, shadow 1's color (0 <= r, g, b <= 1) + 102/255), per r, g, b component dir = shadow 1's direction + 180°
shdw18 (3D Inner Box Shadow)	 Equivalent to two outer shadow effects. Shadow 1: No additional attributes specified. Shadow 2: color = min(1, shadow 1's color (0 <= r, g, b <= 1) + 102/255), per r, g, b component dir = shadow 1's direction + 180°
shdw19 (Back Center Perspective Shadow)	 align = "b" sy = 50°
shdw2 (Top Right Drop Shadow)	 No additional attributes specified.
shdw20 (Front Bottom Shadow)	 align = "b" sy = -100°
shdw3 (Back Left Perspective Shadow)	 align = "b" ky = 40.89° sy = 50%
shdw4 (Back Right Perspective Shadow)	 align = "b" kx = -40.89° sy = 50%
shdw5 (Bottom Left Drop Shadow)	 No additional attributes specified.
shdw6 (Bottom Right Drop Shadow)	 No additional attributes specified.
shdw7 (Front Left Perspective Shadow)	

Enumeration Value	Description
	align = "b" kx = 40.89° sy = -50%
shdw8 (Front Right Perspective Shadow)	 align = "b" kx = -40.89° sy = -50%
shdw9 (Top Left Small Drop Shadow)	 align = "tl" sx = 75% sy = 75%

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PresetShadowVal) is located in §A.4.1. *end note*]

20.1.10.52 ST_RectAlignment (Rectangle Alignments)

This simple type describes how to position two rectangles relative to each other.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Rectangle Alignment Enum (Bottom))	Bottom
bl (Rectangle Alignment Enum (Bottom Left))	Bottom Left
br (Rectangle Alignment Enum (Bottom Right))	Bottom Right
ctr (Rectangle Alignment Enum (Center))	Center
l (Rectangle Alignment Enum (Left))	Left
r (Rectangle Alignment Enum (Right))	Right
t (Rectangle Alignment Enum (Top))	Top
tl (Rectangle Alignment Enum (Top Left))	Top Left
tr (Rectangle Alignment Enum (Top Right))	Top Right

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RectAlignment) is located in §A.4.1. *end note*]

20.1.10.53 ST_SchemeColorVal (Scheme Color)

This simple type represents a scheme color value.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
accent1 (Accent Color 1)	Extra scheme color 1
accent2 (Accent Color 2)	Extra scheme color 2
accent3 (Accent Color 3)	Extra scheme color 3
accent4 (Accent Color 4)	Extra scheme color 4
accent5 (Accent Color 5)	Extra scheme color 5
accent6 (Accent Color 6)	Extra scheme color 6
bg1 (Background Color 1)	Semantic background color
bg2 (Background Color 2)	Semantic additional background color
dk1 (Dark Color 1)	Main dark color 1
dk2 (Dark Color 2)	Main dark color 2
folHlink (Followed Hyperlink Color)	Followed Hyperlink Color
hlink (Hyperlink Color)	Regular Hyperlink Color
lt1 (Light Color 1)	Main Light Color 1
lt2 (Light Color 2)	Main Light Color 2
phClr (Style Color)	A color used in theme definitions which means to use the color of the style.
tx1 (Text Color 1)	Semantic text color
tx2 (Text Color 2)	Semantic additional text color

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SchemeColorVal) is located in §A.4.1. *end note*]

20.1.10.54 ST_ShapeID (Shape ID)

Specifies the shape ID for legacy shape identification purposes.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ShapeID) is located in §A.4.1. *end note*]

20.1.10.55 ST_ShapeType (Preset Shape Types)

This simple type specifies the preset shape geometry that is to be used for a shape. An enumeration of this simple type is used so that a custom geometry does not have to be specified but instead can be constructed automatically by the generating application. For each enumeration listed there is also the corresponding DrawingML code that would be used to construct this shape were it a custom geometry. Within the construction

code for each of these preset shapes there are predefined guides that the generating application shall maintain for calculation purposes at all times. The necessary guides should have the following values.

3/4 of a Circle ('3cd4') - Constant value of "16200000.0"

The units here are in 60,000ths of a degree. This is equivalent to 270 degrees.

3/8 of a Circle ('3cd8') - Constant value of "8100000.0"

The units here are in 60,000ths of a degree. This is equivalent to 135 degrees.

5/8 of a Circle ('5cd8') - Constant value of "13500000.0"

The units here are in 60,000ths of a degree. This is equivalent to 225 degrees.

7/8 of a Circle ('7cd8') - Constant value of "18900000.0"

The units here are in 60,000ths of a degree. This is equivalent to 315 degrees.

Shape Bottom Edge ('b') - Constant value of "h"

This is the bottom edge of the shape and since the top edge of the shape is considered the 0 point, the bottom edge is thus the shape height.

1/2 of a Circle ('cd2') - Constant value of "10800000.0"

The units here are in 60,000ths of a degree. This is equivalent to 180 degrees.

1/4 of a Circle ('cd4') - Constant value of "5400000.0"

The units here are in 60,000ths of a degree. This is equivalent to 90 degrees.

1/8 of a Circle ('cd8') - Constant value of "2700000.0"

The units here are in 60,000ths of a degree. This is equivalent to 45 degrees.

Shape Height ('h')

This is the variable height of the shape defined in the shape properties. This value is received from the shape transform listed within the <spPr> element.

Horizontal Center ('hc') - Calculated value of "*/ w 1.0 2.0"

This is the horizontal center of the shape which is just the width divided by 2.

1/2 of Shape Height ('hd2') - Calculated value of "*/ h 1.0 2.0"

This is 1/2 the shape height.

1/4 of Shape Height ('hd4') - Calculated value of "*/ h 1.0 4.0"

This is 1/4 the shape height.

1/5 of Shape Height ('hd5') - Calculated value of "*/ h 1.0 5.0"

This is 1/5 the shape height.

1/6 of Shape Height ('hd6') - Calculated value of "*/ h 1.0 6.0"

This is 1/6 the shape height.

1/8 of Shape Height ('hd8') - Calculated value of "*/ h 1.0 8.0"

This is 1/8 the shape height.

Shape Left Edge ('l') - Constant value of "0"

This is the left edge of the shape and the left edge of the shape is considered the horizontal 0 point.

Longest Side of Shape ('ls') - Calculated value of "max w h"

This is the longest side of the shape. This value is either the width or the height depending on which is greater.

Shape Right Edge ('r') - Constant value of "w"

This is the right edge of the shape and since the left edge of the shape is considered the 0 point, the right edge is thus the shape width.

Shortest Side of Shape ('ss') - Calculated value of "min w h"

This is the shortest side of the shape. This value is either the width or the height depending on which is smaller.

1/2 Shortest Side of Shape ('ssd2') - Calculated value of "*/ ss 1.0 2.0"

This is 1/2 the shortest side of the shape.

1/4 Shortest Side of Shape ('ssd4') - Calculated value of "*/ ss 1.0 4.0"

This is 1/4 the shortest side of the shape.

1/6 Shortest Side of Shape ('ssd6') - Calculated value of "*/ ss 1.0 6.0"

This is 1/6 the shortest side of the shape.

1/8 Shortest Side of Shape ('ssd8') - Calculated value of "*/ ss 1.0 8.0"

This is 1/8 the shortest side of the shape.

Shape Top Edge ('t') - Constant value of "0"

This is the top edge of the shape and the top edge of the shape is considered the vertical 0 point.

Vertical Center of Shape ('vc') - Calculated value of "*/ h 1.0 2.0"

This is the vertical center of the shape which is just the height divided by 2.

Shape Width ('w')

This is the variable width of the shape defined in the shape properties. This value is received from the shape transform listed within the <spPr> element.

1/2 of Shape Width ('wd2') - Calculated value of "*/ w 1.0 2.0"

This is 1/2 the shape width.

1/4 of Shape Width ('wd4') - Calculated value of "*/ w 1.0 4.0"

This is 1/4 the shape width.

1/5 of Shape Width ('wd5') - Calculated value of "*/ w 1.0 5.0"

This is 1/5 the shape width.

1/6 of Shape Width ('wd6') - Calculated value of "*/ w 1.0 6.0"

This is 1/6 the shape width.

1/8 of Shape Width ('wd8') - Calculated value of "*/ w 1.0 8.0"

This is 1/8 the shape width.

1/10 of Shape Width ('wd10') - Calculated value of "*/ w 1.0 10.0"

This is 1/10 the shape width.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
accentBorderCallout1 (Callout 1 with Border and Accent Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the accentBorderCallout1 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
accentBorderCallout2 (Callout 2 with Border and Accent Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the accentBorderCallout2 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
accentBorderCallout3 (Callout 3 with Border and Accent Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the accentBorderCallout3 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

Enumeration Value	Description
	
accentCallout1 (Callout 1 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the accentCallout1 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
accentCallout2 (Callout 2 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the accentCallout2 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
accentCallout3 (Callout 3 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the accentCallout3 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

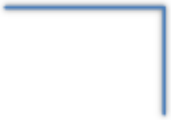
Enumeration Value	Description
	
actionButtonBackPrevious (Back or Previous Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonBackPrevious element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
actionButtonBeginning (Beginning Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonBeginning element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
actionButtonBlank (Blank Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in

Enumeration Value	Description
	the actionButtonBlank element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
actionButtonDocument (Document Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonDocument element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
actionButtonEnd (End Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonEnd element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 

Enumeration Value	Description
actionButtonForwardNext (Forward or Next Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonForwardNext element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
actionButtonHelp (Help Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonHelp element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
actionButtonHome (Home Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonHome element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

Enumeration Value	Description
	
actionButtonInformation (Information Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonInformation element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
actionButtonMovie (Movie Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonMovie element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
actionButtonReturn (Return Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to

Enumeration Value	Description
	generate this preset shape definition is contained in the actionButtonReturn element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
actionButtonSound (Sound Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonSound element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
arc (Curved Arc Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the arc element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
bentArrow (Bent Arrow Shape)	Specifies a preset shape geometry. This geometry shall

Enumeration Value	Description
	be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the bentArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
bentConnector2 (Bent Connector 2 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the bentConnector2 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
bentConnector3 (Bent Connector 3 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the bentConnector3 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

Enumeration Value	Description
	
bentConnector4 (Bent Connector 4 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the bentConnector4 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
bentConnector5 (Bent Connector 5 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the bentConnector5 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
bentUpArrow (Bent Up Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the bentUpArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are

Enumeration Value	Description
	described in further detail above. *end note*] 
bevel (Bevel Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the bevel element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
blockArc (Block Arc Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the blockArc element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
borderCallout1 (Callout 1 with Border Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the borderCallout1 element in the preset shape geometries electronic addenda of Annex D. The

Enumeration Value	Description
	constants used in that markup are guides that are described in further detail above. *end note*] 
borderCallout2 (Callout 2 with Border Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the borderCallout2 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
borderCallout3 (Callout 3 with Border Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the borderCallout3 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
bracePair (Brace Pair Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the bracePair element in the preset shape geometries

Enumeration Value	Description
	electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
bracketPair (Bracket Pair Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the bracketPair element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
callout1 (Callout 1 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the callout1 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
callout2 (Callout 2 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below.

Enumeration Value	Description
	[*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the callout2 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
callout3 (Callout 3 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the callout3 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
can (Can Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the can element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 

Enumeration Value	Description
chartPlus (Chart Plus Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the chartPlus element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
chartStar (Chart Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the chartStar element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
chartX (Chart X Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the chartX element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 

Enumeration Value	Description
chevron (Chevron Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the chevron element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
chord (Chord Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the chord element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
circularArrow (Circular Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the circularArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 

Enumeration Value	Description
cloud (Cloud Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the cloud element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
cloudCallout (Callout Cloud Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the cloudCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
corner (Corner Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the corner element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

Enumeration Value	Description
	
cornerTabs (Corner Tabs Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the cornerTabs element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
cube (Cube Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the cube element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
curvedConnector2 (Curved Connector 2 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the curvedConnector2 element in the preset shape

Enumeration Value	Description
	geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
curvedConnector3 (Curved Connector 3 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the curvedConnector3 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
curvedConnector4 (Curved Connector 4 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the curvedConnector4 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
curvedConnector5 (Curved Connector 5 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to

Enumeration Value	Description
	generate this preset shape definition is contained in the curvedConnector5 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
curvedDownArrow (Curved Down Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the curvedDownArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
curvedLeftArrow (Curved Left Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the curvedLeftArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
curvedRightArrow (Curved Right Arrow Shape)	Specifies a preset shape geometry. This geometry shall

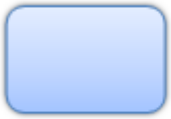
Enumeration Value	Description
	be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the curvedRightArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
curvedUpArrow (Curved Up Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the curvedUpArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
decagon (Decagon Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the decagon element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

Enumeration Value	Description
	
diagStripe (Diagonal Stripe Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the diagStripe element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
diamond (Diamond Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the diamond element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
dodecagon (Dodecagon Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the dodecagon element in the preset shape

Enumeration Value	Description
	geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
donut (Donut Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the donut element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
doubleWave (Double Wave Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the doubleWave element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
downArrow (Down Arrow Shape)	Specifies a preset shape geometry. This geometry shall

Enumeration Value	Description
	be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the downArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
downArrowCallout (Callout Down Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the downArrowCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
ellipse (Ellipse Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the ellipse element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

Enumeration Value	Description
	
ellipseRibbon (Ellipse Ribbon Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the ellipseRibbon element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
ellipseRibbon2 (Ellipse Ribbon 2 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the ellipseRibbon2 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartAlternateProcess (Alternate Process Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartAlternateProcess element in the preset shape geometries electronic addenda of Annex D. The

Enumeration Value	Description
	constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartCollate (Collate Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartCollate element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartConnector (Connector Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartConnector element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartDecision (Decision Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartDecision element in the preset shape geometries electronic addenda of Annex D. The

Enumeration Value	Description
	constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartDelay (Delay Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartDelay element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartDisplay (Display Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartDisplay element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartDocument (Document Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartDocument element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are

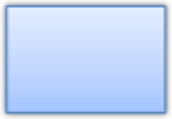
Enumeration Value	Description
	described in further detail above. *end note*] 
flowChartExtract (Extract Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartExtract element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartInputOutput (Input Output Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartInputOutput element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartInternalStorage (Internal Storage Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartInternalStorage element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are

Enumeration Value	Description
	described in further detail above. *end note*] 
flowChartMagneticDisk (Magnetic Disk Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartMegneticDisk element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartMagneticDrum (Magnetic Drum Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartMagneticDrum element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartMagneticTape (Magnetic Tape Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartMagneticTape element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are

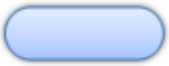
Enumeration Value	Description
	described in further detail above. *end note*] 
flowChartManualInput (Manual Input Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartManualInput element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartManualOperation (Manual Operation Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartManualOperation element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartMerge (Merge Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartMerge element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

Enumeration Value	Description
	
flowChartMultidocument (Multi-Document Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartMultidocument element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartOfflineStorage (Offline Storage Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartOfflineStorage element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartOffpageConnector (Off-Page Connector Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartOffpageConnector element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that

Enumeration Value	Description
	are described in further detail above. *end note*] 
flowChartOnlineStorage (Online Storage Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartOnlineStorage element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartOr (Or Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartOr element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartPredefinedProcess (Predefined Process Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartPredefinedProcess element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that

Enumeration Value	Description
	are described in further detail above. *end note*] 
flowChartPreparation (Preparation Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartPreparation element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartProcess (Process Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartProcess element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartPunchedCard (Punched Card Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartPunchedCard element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are

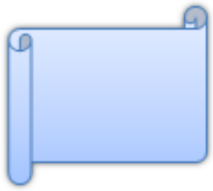
Enumeration Value	Description
	described in further detail above. *end note*] 
flowChartPunchedTape (Punched Tape Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartPunchedTape element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartSort (Sort Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartSort element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartSummingJunction (Summing Junction Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in

Enumeration Value	Description
	the flowChartSummingJunction element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartTerminator (Terminator Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartTerminator element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
foldedCorner (Folded Corner Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the foldedCorner element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
frame (Frame Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the frame element in the preset shape geometries

Enumeration Value	Description
	electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
funnel (Funnel Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the funnel element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
gear6 (Gear 6 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the gear6 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
gear9 (Gear 9 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in

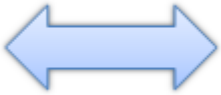
Enumeration Value	Description
	the gear9 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
halfFrame (Half Frame Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the halfFrame element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
heart (Heart Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the heart element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
heptagon (Heptagon Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below.

Enumeration Value	Description
	[*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the heptagon element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
hexagon (Hexagon Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the hexagon element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
homePlate (Home Plate Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the homePlate element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 

Enumeration Value	Description
horizontalScroll (Horizontal Scroll Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the horizontalScroll element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
irregularSeal1 (Irregular Seal 1 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the irregularSeal1 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
irregularSeal2 (Irregular Seal 2 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the irregularSeal2 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

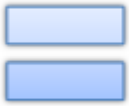
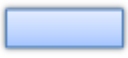
Enumeration Value	Description
	
leftArrow (Left Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
leftArrowCallout (Callout Left Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftArrowCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
leftBrace (Left Brace Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftBrace element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

Enumeration Value	Description
	
leftBracket (Left Bracket Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftBracket element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
leftCircularArrow (Left Circular Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftCircularArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
leftRightArrow (Left Right Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftRightArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are

Enumeration Value	Description
	described in further detail above. *end note*] 
leftRightArrowCallout (Callout Left Right Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftRightArrowCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
leftRightCircularArrow (Left Right Circular Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftRightCircularArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
leftRightRibbon (Left Right Ribbon Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftRightRibbon element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

Enumeration Value	Description
	
leftRightUpArrow (Left Right Up Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftRightUpArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
leftUpArrow (Left Up Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftUpArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
lightningBolt (Lightning Bolt Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the lightningBolt element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

Enumeration Value	Description
	
line (Line Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the line element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
lineInv (Line Inverse Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the lineInv element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
mathDivide (Divide Math Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in

Enumeration Value	Description
	the mathDivide element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
mathEqual (Equal Math Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the mathEqual element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
mathMinus (Minus Math Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the mathMinus element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
mathMultiply (Multiply Math Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the mathMultiply element in the preset shape geometries electronic addenda of Annex D. The

Enumeration Value	Description
	constants used in that markup are guides that are described in further detail above. *end note*] 
mathNotEqual (Not Equal Math Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the mathNotEqual element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
mathPlus (Plus Math Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the mathPlus element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
moon (Moon Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the moon element in the preset shape geometries

Enumeration Value	Description
	electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
nonIsoscelesTrapezoid (Non-Isosceles Trapezoid Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the nonIsocelesTrapezoid element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
noSmoking (No Smoking Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the noSmoking element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
notchedRightArrow (Notched Right Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to

Enumeration Value	Description
	generate this preset shape definition is contained in the notchedRightArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
octagon (Octagon Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the octagon element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
parallelogram (Parallelogram Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the parallelogram element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
pentagon (Pentagon Shape)	Specifies a preset shape geometry. This geometry shall

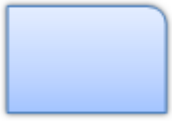
Enumeration Value	Description
	be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the pentagon element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
pie (Pie Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the pie element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
pieWedge (Pie Wedge Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the pieWedge element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

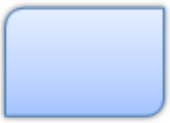
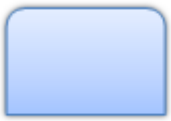
Enumeration Value	Description
	
plaque (Plaque Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the plaque element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
plaqueTabs (Plaque Tabs Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the plaqueTabs element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
plus (Plus Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the plus element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

Enumeration Value	Description
	
quadArrow (Quad-Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the quadArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
quadArrowCallout (Callout Quad-Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the quadArrowCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
rect (Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below.

Enumeration Value	Description
	[*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the rect element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
ribbon (Ribbon Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the ribbon element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
ribbon2 (Ribbon 2 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the ribbon2 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
rightArrow (Right Arrow Shape)	Specifies a preset shape geometry. This geometry shall

Enumeration Value	Description
	be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the rightArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
rightArrowCallout (Callout Right Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the rightArrowCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
rightBrace (Right Brace Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the rightBrace element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

Enumeration Value	Description
	
rightBracket (Right Bracket Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the rightBracket element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
round1Rect (One Round Corner Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the round1Rect element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
round2DiagRect (Two Diagonal Round Corner Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the round2DiagRect element in the preset shape geometries electronic addenda of Annex D. The

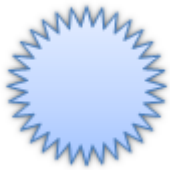
Enumeration Value	Description
	constants used in that markup are guides that are described in further detail above. *end note*] 
round2SameRect (Two Same-side Round Corner Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the round2SameRect element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
roundRect (Round Corner Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the roundRect element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
rtTriangle (Right Triangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the rtTriangle element in the preset shape geometries

Enumeration Value	Description
	electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
smileyFace (Smiley Face Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the smileyFace element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
snip1Rect (One Snip Corner Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the snip1Rect element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
snip2DiagRect (Two Diagonal Snip Corner Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below.

Enumeration Value	Description
	[*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the snip2DiagRect element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
snip2SameRect (Two Same-side Snip Corner Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the snip2SameRect element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
snipRoundRect (One Snip One Round Corner Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the snipRoundRect element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 

Enumeration Value	Description
squareTabs (Square Tabs Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the squareTabs element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
star10 (Ten Pointed Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the star10 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
star12 (Twelve Pointed Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the star12 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

Enumeration Value	Description
	
star16 (Sixteen Pointed Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the star16 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
star24 (Twenty Four Pointed Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the star24 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
star32 (Thirty Two Pointed Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the star32 element in the preset shape geometries electronic addenda of Annex D. The constants used in

Enumeration Value	Description
	that markup are guides that are described in further detail above. *end note*] 
star4 (Four Pointed Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the star4 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
star5 (Five Pointed Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the star5 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
star6 (Six Pointed Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below.

Enumeration Value	Description
	[*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the star6 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
star7 (Seven Pointed Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the star7 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
star8 (Eight Pointed Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the star8 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 

Enumeration Value	Description
straightConnector1 (Straight Connector 1 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the straightConnector1 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
stripedRightArrow (Striped Right Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the stripedRightArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
sun (Sun Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the sun element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

Enumeration Value	Description
	
swooshArrow (Swoosh Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the swooshArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
teardrop (Teardrop Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the teardrop element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
trapezoid (Trapezoid Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the trapezoid element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further

Enumeration Value	Description
	detail above. *end note*] 
triangle (Triangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the triangle element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
upArrow (Up Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the upArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
upArrowCallout (Callout Up Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below.

Enumeration Value	Description
	[*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the upArrowCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
upDownArrow (Up Down Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the upDownArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
upDownArrowCallout (Callout Up Down Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the upDownArrowCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

Enumeration Value	Description
	
uturnArrow (U-Turn Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the uturnArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
verticalScroll (Vertical Scroll Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the verticalScroll element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
wave (Wave Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the wave element in the preset shape geometries

Enumeration Value	Description
	electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
wedgeEllipseCallout (Callout Wedge Ellipse Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the wedgeEllipseCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
wedgeRectCallout (Callout Wedge Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the wedgeRectCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
wedgeRoundRectCallout (Callout Wedge Round Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to

Enumeration Value	Description
	generate this preset shape definition is contained in the wedgeRoundRectCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ShapeType) is located in §A.4.1. *end note*]

20.1.10.56 ST_StyleMatrixColumnIndex (Style Matrix Column Index)

This simple type specifies an index into one of the lists in the style matrix specified by the fmtScheme element (bgFillStyleLst, effectStyleLst, fillStyleLst, or lnStyleLst).

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_StyleMatrixColumnIndex) is located in §A.4.1. *end note*]

20.1.10.57 ST_SystemColorVal (System Color Value)

This simple type specifies a system color value. This color is based upon the value that this color currently has within the system on which the document is being viewed.

Applications shall use the lastClr attribute to determine the absolute value of the last color used if system colors are not supported.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
3dDkShadow (3D Dark System Color)	Specifies a Dark shadow color for three-dimensional display elements.
3dLight (3D Light System Color)	Specifies a Light color for three-dimensional display elements (for edges facing the light source).
activeBorder (Active Border System Color)	Specifies an Active Window Border Color.
activeCaption (Active Caption System Color)	Specifies the active window title bar color. In particular the left side color in the color gradient of an active

Enumeration Value	Description
	window's title bar if the gradient effect is enabled.
appWorkspace (Application Workspace System Color)	Specifies the Background color of multiple document interface (MDI) applications.
background (Background System Color)	Specifies the desktop background color.
btnFace (Button Face System Color)	Specifies the face color for three-dimensional display elements and for dialog box backgrounds.
btnHighlight (Button Highlight System Color)	Specifies the highlight color for three-dimensional display elements (for edges facing the light source).
btnShadow (Button Shadow System Color)	Specifies the shadow color for three-dimensional display elements (for edges facing away from the light source).
btnText (Button Text System Color)	Specifies the color of text on push buttons.
captionText (Caption Text System Color)	Specifies the color of text in the caption, size box, and scroll bar arrow box.
gradientActiveCaption (Gradient Active Caption System Color)	Specifies the right side color in the color gradient of an active window's title bar.
gradientInactiveCaption (Gradient Inactive Caption System Color)	Specifies the right side color in the color gradient of an inactive window's title bar.
grayText (Gray Text System Color)	Specifies a grayed (disabled) text. This color is set to 0 if the current display driver does not support a solid gray color.
highlight (Highlight System Color)	Specifies the color of Item(s) selected in a control.
highlightText (Highlight Text System Color)	Specifies the text color of item(s) selected in a control.
hotLight (Hot Light System Color)	Specifies the color for a hyperlink or hot-tracked item.
inactiveBorder (Inactive Border System Color)	Specifies the color of the Inactive window border.
inactiveCaption (Inactive Caption System Color)	Specifies the color of the Inactive window caption. Specifies the left side color in the color gradient of an inactive window's title bar if the gradient effect is enabled.
inactiveCaptionText (Inactive Caption Text System Color)	Specifies the color of text in an inactive caption.
infoBk (Info Back System Color)	Specifies the background color for tooltip controls.
infoText (Info Text System Color)	Specifies the text color for tooltip controls.
menu (Menu System Color)	Specifies the menu background color.
menuBar (Menu Bar System Color)	Specifies the background color for the menu bar when menus appear as flat menus.
menuHighlight (Menu Highlight System Color)	Specifies the color used to highlight menu items when the menu appears as a flat menu.

Enumeration Value	Description
menuText (Menu Text System Color)	Specifies the color of Text in menus.
scrollBar (Scroll Bar System Color)	Specifies the scroll bar gray area color.
window (Window System Color)	Specifies window background color.
windowFrame (Window Frame System Color)	Specifies the window frame color.
windowText (Window Text System Color)	Specifies the color of text in windows.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SystemColorVal) is located in §A.4.1. *end note*]

20.1.10.58 ST_TextAlignType (Text Alignment Types)

This simple type specifies the text alignment types

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ctr (Text Alignment Enum (Center))	Align text in the center.
dist (Text Alignment Enum (Distributed))	Distributes the text words across an entire text line.
just (Text Alignment Enum (Justified))	Align text so that it is justified across the whole line. It is smart in the sense that it does not justify sentences which are short.
justLow (Text Alignment Enum (Justified Low))	Aligns the text with an adjusted kashida length for Arabic text.
l (Text Alignment Enum (Left))	Align text to the left margin.
r (Text Alignment Enum (Right))	Align text to the right margin.
thaiDist (Text Alignment Enum (Thai Distributed))	Distributes Thai text specially, because each character is treated as a word.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextAlignType) is located in §A.4.1. *end note*]

20.1.10.59 ST_TextAnchoringType (Text Anchoring Types)

This simple type specifies a list of available anchoring types for text.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Text Anchor Enum (Bottom))	Anchor the text at the bottom of the bounding rectangle.
ctr (Text Anchor Enum (Center))	Anchor the text at the middle of the bounding rectangle.
dist (Text Anchor Enum (Distributed))	Anchor the text so that it is distributed vertically. When text is horizontal, this spaces out the actual lines of text and is almost always identical in behavior to anchorJustified (special case: if only 1 line, then anchored in middle). When text is vertical, then it distributes the letters vertically. This is different than anchorJustified, because it always forces distribution of the words, even if there are only one or two words in a line.
just (Text Anchor Enum (Justified))	Anchor the text so that it is justified vertically. When text is horizontal, this spaces out the actual lines of text and is almost always identical in behavior to 'distrib' (special case: if only 1 line, then anchored at top). When text is vertical, then it justifies the letters vertically. This is different than anchorDistributed, because in some cases such as very little text in a line, it does not justify.
t (Text Anchoring Type Enum (Top))	Anchor the text at the top of the bounding rectangle.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextAnchoringType) is located in §A.4.1. *end note*]

20.1.10.60 ST_TextAutonumberScheme (Text Auto-number Schemes)

This simple type specifies a list of automatic numbering schemes.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
alphaLcParenBoth (Autonumber Enum (alphaLcParenBoth))	(a), (b), (c), …
alphaLcParenR (Autonumbering Enum (alphaLcParenR))	a), b), c), …
alphaLcPeriod (Autonumbering Enum (alphaLcPeriod))	a., b., c., …
alphaUcParenBoth (Autonumbering Enum (	(A), (B), (C), …

Enumeration Value	Description
alphaUcParenBoth))	
alphaUcParenR (Autonumbering Enum (alphaUcParenR))	A), B), C), …
alphaUcPeriod (Autonumbering Enum (alphaUcPeriod))	A., B., C., …
arabic1Minus (Autonumbering Enum (arabic1Minus))	Bidi Arabic 1 (AraAlpha) with ANSI minus symbol
arabic2Minus (Autonumbering Enum (arabic2Minus))	Bidi Arabic 2 (AraAbjad) with ANSI minus symbol
arabicDbPeriod (Autonumbering Enum (arabicDbPeriod))	Dbl-byte Arabic numbers w/ double-byte period
arabicDbPlain (Autonumbering Enum (arabicDbPlain))	Dbl-byte Arabic numbers
arabicParenBoth (Autonumbering Enum (arabicParenBoth))	(1), (2), (3), …
arabicParenR (Autonumbering Enum (arabicParenR))	1), 2), 3), …
arabicPeriod (Autonumbering Enum (arabicPeriod))	1., 2., 3., …
arabicPlain (Autonumbering Enum (arabicPlain))	1, 2, 3, …
circleNumDbPlain (Autonumbering Enum (circleNumDbPlain))	Dbl-byte circle numbers (1-10 circle[0x2460-], 11-arabic numbers)
circleNumWdBlackPlain (Autonumbering Enum (circleNumWdBlackPlain))	Wingdings black circle numbers
circleNumWdWhitePlain (Autonumbering Enum (circleNumWdWhitePlain))	Wingdings white circle numbers (0-10 circle[0x0080-], 11- arabic numbers)
ea1ChsPeriod (Autonumbering Enum (ea1ChsPeriod))	EA: Simplified Chinese w/ single-byte period
ea1ChsPlain (Autonumbering Enum (ea1ChsPlain))	EA: Simplified Chinese (TypeA 1-99, TypeC 100-)
ea1ChtPeriod (Autonumbering Enum (ea1ChtPeriod))	EA: Traditional Chinese w/ single-byte period
ea1ChtPlain (Autonumbering Enum (ea1ChtPlain))	EA: Traditional Chinese (TypeA 1-19, TypeC 20-)
ea1JpnChsDbPeriod (Autonumbering Enum (ea1JpnChsDbPeriod))	EA: Japanese w/ double-byte period
ea1JpnKorPeriod (Autonumbering Enum (ea1JpnKorPeriod))	EA: Japanese/Korean w/ single-byte period
ea1JpnKorPlain (Autonumbering Enum (ea1JpnKorPlain))	EA: Japanese/Korean (TypeC 1-)
hebrew2Minus (Autonumbering Enum (hebrew2Minus))	Bidi Hebrew 2 with ANSI minus symbol

Enumeration Value	Description
hindiAlpha1Period (Autonumbering Enum (hindiAlpha1Period))	Hindi alphabet period - consonants
hindiAlphaPeriod (Autonumbering Enum (hindiAlphaPeriod))	Hindi alphabet period - vowels
hindiNumParenR (Autonumbering Enum (hindiNumParenR))	Hindi numerical parentheses - right
hindiNumPeriod (Autonumbering Enum (hindiNumPeriod))	Hindi numerical period
romanLcParenBoth (Autonumbering Enum (romanLcParenBoth))	(i), (ii), (iii), …
romanLcParenR (Autonumbering Enum (romanLcParenR))	i), ii), iii), …
romanLcPeriod (Autonumbering Enum (romanLcPeriod))	i., ii., iii., …
romanUcParenBoth (Autonumbering Enum (romanUcParenBoth))	(I), (II), (III), …
romanUcParenR (Autonumbering Enum (romanUcParenR))	I), II), III), …
romanUcPeriod (Autonumbering Enum (romanUcPeriod))	I., II., III., …
thaiAlphaParenBoth (Autonumbering Enum (thaiAlphaParenBoth))	Thai alphabet parentheses - both
thaiAlphaParenR (Autonumbering Enum (thaiAlphaParenR))	Thai alphabet parentheses - right
thaiAlphaPeriod (Autonumbering Enum (thaiAlphaPeriod))	Thai alphabet period
thaiNumParenBoth (Autonumbering Enum (thaiNumParenBoth))	Thai numerical parentheses - both
thaiNumParenR (Autonumbering Enum (thaiNumParenR))	Thai numerical parentheses - right
thaiNumPeriod (Autonumbering Enum (thaiNumPeriod))	Thai numerical period

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextAutonumberScheme) is located in §A.4.1. *end note*]

20.1.10.61 ST_TextBulletSizePercent (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.

This simple type also specifies the following restrictions:

- This simple type's contents shall match the following regular expression pattern: 0*((2[5-9])|([3-9][0-9])|([1-3][0-9][0-9])|400)%.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextBulletSizePercent) is located in §A.4.1. *end note*]

20.1.10.62 ST_TextBulletStartAtNum (Start Bullet At Number)

This simple type specifies the range that the start at number for a bullet's auto-numbering sequence can begin at. When the numbering is alphabetical, then the numbers map to the appropriate letter. 1->a, 2->b, etc. If the numbers go above 26, then the numbers begin to double up. For example, 27->aa and 53->aaa.

This simple type's contents are a restriction of the W3C XML Schema int datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 1.
- This simple type has a maximum value of less than or equal to 32767.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextBulletStartAtNum) is located in §A.4.1. *end note*]

20.1.10.63 ST_TextCapsType (Text Cap Types)

This simple type specifies the cap types of the text.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
all (Text Caps Enum (All))	Apply all caps on the text. All lower case letters are converted to upper case even though they are stored differently in the backing store.
none (Text Caps Enum (None))	The reason we cannot implicitly have noCaps be the scenario where capitalization is not specified is because not being specified implies deriving from a particular style and the user might want to override that and make some text not have a capitalization scheme even though the style says otherwise.
small (Text Caps Enum (Small))	Apply small caps to the text. All letters are converted to lower case.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextCapsType) is located in §A.4.1. *end note*]

20.1.10.64 ST_TextColumnCount (Text Column Count)

This simple type specifies the number of columns.

This simple type's contents are a restriction of the W3C XML Schema int datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 1.
- This simple type has a maximum value of less than or equal to 16.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextColumnCount) is located in §A.4.1. *end note*]

20.1.10.65 ST_TextFontAlignType (Font Alignment Types)

This simple type specifies the different kinds of font alignment.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
auto (Font Alignment Enum (Automatic))	When the text flow is horizontal or simple vertical same as fontBaseline but for other vertical modes same as fontCenter.
b (Font Alignment Enum (Bottom))	The letters are anchored to the very bottom of a single line. This is different than the bottom baseline because of letters such as "g," "q," "y," etc.
base (Font Alignment Enum (Baseline))	The letters are anchored to the bottom baseline of a single line.
ctr (Font Alignment Enum (Center))	The letters are anchored between the two baselines of a single line.
t (Font Alignment Enum (Top))	The letters are anchored to the top baseline of a single line.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextFontAlignType) is located in §A.4.1. *end note*]

20.1.10.66 ST_TextFontScalePercentOrPercentString (Text Font Scale Percentage)

This simple type specifies that its contents will contain a text font scale percent percentage. See the union's member types for details.

This simple type is a union of the following types:

- The ST_Percentage simple type (§22.9.2.9).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextFontScalePercentOrPercentString) is located in §A.4.1. *end note*]

20.1.10.67 ST_TextFontSize (Text Font Size)

This simple type specifies the size of any text in hundredths of a point. Shall be at least 1 point.

This simple type's contents are a restriction of the W3C XML Schema int 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 400000.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextFontSize) is located in §A.4.1. *end note*]

20.1.10.68 ST_TextHorzOverflowType (Text Horizontal Overflow Types)

This simple type specifies the text horizontal overflow types

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
clip (Text Horizontal Overflow Enum (Clip))	When a big character does not fit into a line, clip it at the proper horizontal overflow.
overflow (Text Horizontal Overflow Enum (Overflow))	When a big character does not fit into a line, allow a horizontal overflow.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextHorzOverflowType) is located in §A.4.1. *end note*]

20.1.10.69 ST_TextIndent (Text Indentation)

This simple type specifies the text indentation amount to be used.

The units of measurement used here are EMUs (English Metric Units).

This simple type's contents are a restriction of the ST_Coordinate32Unqualified datatype (§20.1.10.18).

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to -51206400.
- This simple type has a maximum value of less than or equal to 51206400.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextIndent) is located in §A.4.1. *end note*]

20.1.10.70 ST_TextIndentLevelType (Text Indent Level Type)

This simple type specifies the indent level type. We support list level 0 to 8, and we use -1 and -2 for outline mode levels that should only exist in memory.

This simple type's contents are a restriction of the W3C XML Schema int 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 8.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextIndentLevelType) is located in §A.4.1. *end note*]

20.1.10.71 ST_TextMargin (Text Margin)

This simple type specifies the margin that is used and its corresponding size.

This simple type's contents are a restriction of the ST_Coordinate32Unqualified datatype (§20.1.10.18).

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 51206400.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextMargin) is located in §A.4.1. *end note*]

20.1.10.72 ST_TextNonNegativePoint (Text Non-Negative Point)

This simple type specifies a non-negative font size in hundredths of a point. This is restricted to the range [0, 400000].

This simple type's contents are a restriction of the W3C XML Schema int 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 400000.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextNonNegativePoint) is located in §A.4.1. *end note*]

20.1.10.73 ST_TextPoint (Text Point)

This simple type specifies a coordinate within the document. This can be used for measurements or spacing; its maximum size is +/- 4000 points.

Its contents can contain either:

- A whole number, whose contents consist of a measurement in hundredths of a point
- A number immediately followed by a unit identifier

This simple type is a union of the following types:

- The ST_TextPointUnqualified simple type (§20.1.10.74).
- The ST_UniversalMeasure simple type (§22.9.2.15).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextPoint) is located in §A.4.1. *end note*]

20.1.10.74 ST_TextPointUnqualified (Text Point)

This simple type specifies a font size in hundredths of a point. This is restricted to the range [-400000, 400000], i.e from -4000 pt to 4000 pt.

This simple type's contents are a restriction of the W3C XML Schema int datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to -400000.
- 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_TextPointUnqualified) is located in §A.4.1. *end note*]

20.1.10.75 ST_TextShapeType (Preset Text Shape Types)

This simple type specifies the preset text shape geometry that is to be used for a shape. An enumeration of this simple type is used so that a custom geometry does not have to be specified but instead can be constructed automatically by the generating application. For each enumeration listed there is also the corresponding DrawingML code that would be used to construct this shape were it a custom geometry. Within the construction code for each of these preset text shapes there are predefined guides that the generating application shall maintain for calculation purposes at all times. The necessary guides should have the following values.

3/4 of a Circle ('3cd4') - Constant value of "16200000.0"

The units here are in 60,000ths of a degree. This is equivalent to 270 degrees.

3/8 of a Circle ('3cd8') - Constant value of "8100000.0"

The units here are in 60,000ths of a degree. This is equivalent to 135 degrees.

5/8 of a Circle ('5cd8') - Constant value of "13500000.0"

The units here are in 60,000ths of a degree. This is equivalent to 225 degrees.

7/8 of a Circle ('7cd8') - Constant value of "18900000.0"

The units here are in 60,000ths of a degree. This is equivalent to 315 degrees.

Shape Bottom Edge ('b') - Constant value of "h"

This is the bottom edge of the shape and since the top edge of the shape is considered the 0 point, the bottom edge is thus the shape height.

1/2 of a Circle ('cd2') - Constant value of "10800000.0"

The units here are in 60,000ths of a degree. This is equivalent to 180 degrees.

1/4 of a Circle ('cd4') - Constant value of "5400000.0"

The units here are in 60,000ths of a degree. This is equivalent to 90 degrees.

1/8 of a Circle ('cd8') - Constant value of "2700000.0"

The units here are in 60,000ths of a degree. This is equivalent to 45 degrees.

Shape Height ('h')

This is the variable height of the shape defined in the shape properties. This value is received from the shape transform listed within the <spPr> element.

Horizontal Center ('hc') - Calculated value of "*/ w 1.0 2.0"

This is the horizontal center of the shape which is just the width divided by 2.

1/2 of Shape Height ('hd2') - Calculated value of "*/ h 1.0 2.0"

This is 1/2 the shape height.

1/4 of Shape Height ('hd4') - Calculated value of "*/ h 1.0 4.0"

This is 1/4 the shape height.

1/5 of Shape Height ('hd5') - Calculated value of "*/ h 1.0 5.0"

This is 1/5 the shape height.

1/6 of Shape Height ('hd6') - Calculated value of "*/ h 1.0 6.0"

This is 1/6 the shape height.

1/8 of Shape Height ('hd8') - Calculated value of "*/ h 1.0 8.0"

This is 1/8 the shape height.

Shape Left Edge ('l') - Constant value of "0"

This is the left edge of the shape and the left edge of the shape is considered the horizontal 0 point.

Longest Side of Shape ('ls') - Calculated value of "max w h"

This is the longest side of the shape. This value is either the width or the height depending on which is greater.

Shape Right Edge ('r') - Constant value of "w"

This is the right edge of the shape and since the left edge of the shape is considered the 0 point, the right edge is thus the shape width.

Shortest Side of Shape ('ss') - Calculated value of "min w h"

This is the shortest side of the shape. This value is either the width or the height depending on which is smaller.

1/2 Shortest Side of Shape ('ssd2') - Calculated value of "*/ ss 1.0 2.0"

This is 1/2 the shortest side of the shape.

1/4 Shortest Side of Shape ('ssd4') - Calculated value of "*/ ss 1.0 4.0"

This is 1/4 the shortest side of the shape.

1/6 Shortest Side of Shape ('ssd6') - Calculated value of "*/ ss 1.0 6.0"

This is 1/6 the shortest side of the shape.

1/8 Shortest Side of Shape ('ssd8') - Calculated value of "*/ ss 1.0 8.0"

This is 1/8 the shortest side of the shape.

Shape Top Edge ('t') - Constant value of "0"

This is the top edge of the shape and the top edge of the shape is considered the vertical 0 point.

Vertical Center of Shape ('vc') - Calculated value of "*/ h 1.0 2.0"

This is the vertical center of the shape which is just the height divided by 2.

Shape Width ('w')

This is the variable width of the shape defined in the shape properties. This value is received from the shape transform listed within the <spPr> element.

1/2 of Shape Width ('wd2') - Calculated value of "*/ w 1.0 2.0"

This is 1/2 the shape width.

1/4 of Shape Width ('wd4') - Calculated value of "*/ w 1.0 4.0"

This is 1/4 the shape width.

1/5 of Shape Width ('wd5') - Calculated value of "*/ w 1.0 5.0"

This is 1/5 the shape width.

1/6 of Shape Width ('wd6') - Calculated value of "*/ w 1.0 6.0"

This is 1/6 the shape width.

1/8 of Shape Width ('wd8') - Calculated value of "*/ w 1.0 8.0"

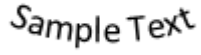
This is 1/8 the shape width.

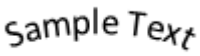
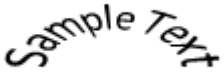
1/10 of Shape Width ('wd10') - Calculated value of "*/ w 1.0 10.0"

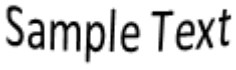
This is 1/10 the shape width.

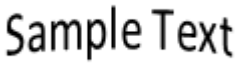
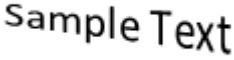
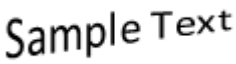
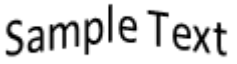
This simple type's contents are a restriction of the W3C XML Schema token datatype.

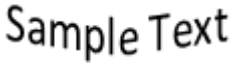
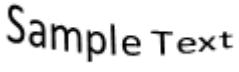
This simple type is restricted to the values listed in the following table:

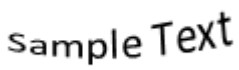
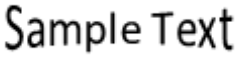
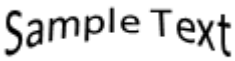
Enumeration Value	Description
textArchDown (Downward Arch Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can

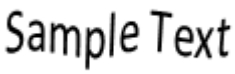
Enumeration Value	Description
	be used to achieve this effect is contained in the textArchDown element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textArchDownPour (Downward Pour Arch Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textArchDownPour element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textArchUp (Upward Arch Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textArchUp element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textArchUpPour (Upward Pour Arch Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textArchUpPour element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

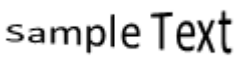
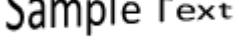
Enumeration Value	Description
textButton (Button Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textButton element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textButtonPour (Button Pour Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textButtonPour element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textCanDown (Downward Can Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textCanDown element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

Enumeration Value	Description
textCanUp (Upward Can Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textCanUp element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textCascadeDown (Downward Cascade Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textCascadeDown element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textCascadeUp (Upward Cascade Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textCascadeUp element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textChevron (Chevron Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the

Enumeration Value	Description
	textChevron element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textChevronInverted (Inverted Chevron Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textChevronInverted element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textCircle (Circle Text Shape)	Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textCircle element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textCirclePour (Circle Pour Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textCirclePour element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textCurveDown (Downward Curve Text Shape)	

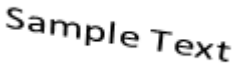
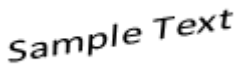
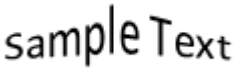
Enumeration Value	Description
	Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textCurveDown element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textCurveUp (Upward Curve Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textCurveUp element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textDeflate (Deflate Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textDeflate element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textDeflateBottom (Bottom Deflate Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textDeflateBottom element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further

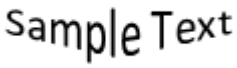
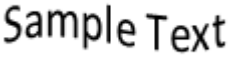
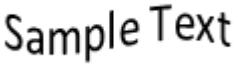
Enumeration Value	Description
	detail above. *end note*]
textDeflateInflate (Deflate-Inflate Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textDeflateInflate element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textDeflateInflateDeflate (Deflate-Inflate-Deflate Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textDeflateInflateDeflate element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textDeflateTop (Top Deflate Text Shape)	Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textDeflateTop element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textDoubleWave1 (Double Wave 1 Text Shape)	 Specifies a text shape that shall match the normative shape shown above.

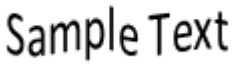
Enumeration Value	Description
	[*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textDoubleWave1 element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textFadeDown (Downward Fade Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textFadeDown element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textFadeLeft (Left Fade Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textFadeLeft element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textFadeRight (Right Fade Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textFadeRight element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

Enumeration Value	Description
textFadeUp (Upward Fade Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textFadeUp element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textInflate (Inflate Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textInflate element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textInflateBottom (Bottom Inflate Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textInflateBottom element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textInflateTop (Top Inflate Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the

Enumeration Value	Description
	textInflateTop element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textNoShape (No Text Shape)	Specifies that the text has no associated shape with it and thus the text should not be warped but instead be constrained by the normal text bounding box.
textPlain (Plain Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textPlain element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textRingInside (Inside Ring Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textRingInside element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textRingOutside (Outside Ring Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textRingOutside element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are

Enumeration Value	Description
	described in further detail above. *end note*]
textSlantDown (Downward Slant Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textSlantDown element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textSlantUp (Upward Slant Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textSlantUp element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textStop (Stop Sign Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textStop element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textTriangle (Triangle Text Shape)	 Specifies a text shape that shall match the normative shape shown above.

Enumeration Value	Description
	[*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textTriangle element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textTriangleInverted (Inverted Triangle Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textTriangleInverted element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textWave1 (Wave 1 Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textWave1 element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textWave2 (Wave 2 Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textWave2 element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

Enumeration Value	Description
textWave4 (Wave 4 Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textWave4 element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextShapeType) is located in §A.4.1. *end note*]

20.1.10.76 ST_TextSpacingPercentOrPercentString (Text Spacing Percent)

This simple type specifies that its contents will contain a text font spacing percentage. See the union's member types for details.

This simple type is a union of the following types:

- The ST_Percentage simple type (§22.9.2.9).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextSpacingPercentOrPercentString) is located in §A.4.1. *end note*]

20.1.10.77 ST_TextSpacingPoint (Text Spacing Point)

This simple type specifies the Text Spacing that is used in terms of font point size.

This simple type's contents are a restriction of the W3C XML Schema int 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 158400.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextSpacingPoint) is located in §A.4.1. *end note*]

20.1.10.78 ST_TextStrikeType (Text Strike Type)

This simple type specifies the strike type.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
dblStrike (Text Strike Enum (Double Strike))	A double strikethrough applied on the text
noStrike (Text Strike Enum (No Strike))	No strike is applied to the text
sngStrike (Text Strike Enum (Single Strike))	A single strikethrough is applied to the text

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextStrikeType) is located in §A.4.1. *end note*]

20.1.10.79 ST_TextTabAlignType (Text Tab Alignment Types)

This simple type specifies the text tab alignment types.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ctr (Text Tab Alignment Enum (Center))	The text at this tab stop is center aligned.
dec (Text Tab Alignment Enum (Decimal))	At this tab stop, the decimals are lined up. From a user's point of view, the text here behaves as right aligned until the decimal, and then as left aligned after the decimal.
l (Text Tab Alignment Enum (Left))	The text at this tab stop is left aligned.
r (Text Tab Alignment Enum (Right))	The text at this tab stop is right aligned.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextTabAlignType) is located in §A.4.1. *end note*]

20.1.10.80 ST_TextTypeface (Text Typeface)

This simple type specifies the way we represent a font typeface.

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_TextTypeface) is located in §A.4.1. *end note*]

20.1.10.81 ST_TextUnderlineType (Text Underline Types)

This simple type specifies the text underline types that is used.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
dash (Text Underline Enum (Dashed))	Underline the text with a single, dashed line of normal thickness.
dashHeavy (Text Underline Enum (Heavy Dashed))	Underline the text with a single, dashed, thick line.
dashLong (Text Underline Enum (Long Dashed))	Underline the text with a single line consisting of long dashes of normal thickness.
dashLongHeavy (Text Underline Enum (Heavy Long Dashed))	Underline the text with a single line consisting of long, thick dashes.
dbl (Text Underline Enum (Double))	Underline the text with two lines of normal thickness.
dotDash (Text Underline Enum (Dot Dash))	Underline the text with a single line of normal thickness consisting of repeating dots and dashes.
dotDashHeavy (Text Underline Enum (Heavy Dot Dash))	Underline the text with a single, thick line consisting of repeating dots and dashes.
dotDotDash (Text Underline Enum (Dot Dot Dash))	Underline the text with a single line of normal thickness consisting of repeating two dots and dashes.
dotDotDashHeavy (Text Underline Enum (Heavy Dot Dot Dash))	Underline the text with a single, thick line consisting of repeating two dots and dashes.
dotted (Text Underline Enum (Dotted))	Underline the text with a single, dotted line of normal thickness.
dottedHeavy (Text Underline Enum (Heavy Dotted))	Underline the text with a single, thick, dotted line.
heavy (Text Underline Enum (Heavy))	Underline the text with a single, thick line.
none (Text Underline Enum (None))	The reason we cannot implicitly have noUnderline be the scenario where underline is not specified is because not being specified implies deriving from a particular style and the user might want to override that and make some text not be underlined even though the style says otherwise.
sng (Text Underline Enum (Single))	Underline the text with a single line of normal thickness.
wavy (Text Underline Enum (Wavy))	Underline the text with a single wavy line of normal thickness.
wavyDbl (Text Underline Enum (Double Wavy))	Underline the text with two wavy lines of normal thickness.
wavyHeavy (Text Underline Enum (Heavy Wavy))	Underline the text with a single, thick wavy line.
words (Text Underline Enum (Words))	Underline just the words and not the spaces between them.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextUnderlineType) is located in §A.4.1. *end note*]

20.1.10.82 ST_TextVerticalType (Vertical Text Types)

If there is vertical text, determines what kind of vertical text is going to be used.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
eaVert (Vertical Text Type Enum (East Asian Vertical))	A special version of vertical text, where some fonts are displayed as if rotated by 90 degrees while some fonts (mostly East Asian) are displayed vertical.
horz (Vertical Text Type Enum (Horizontal))	Horizontal text. This should be default.
mongolianVert (Vertical Text Type Enum (Mongolian Vertical))	A special version of vertical text, where some fonts are displayed as if rotated by 90 degrees while some fonts (mostly East Asian) are displayed vertical. The difference between this and the eastAsianVertical is the text flows top down then LEFT RIGHT, instead of RIGHT LEFT
vert (Vertical Text Type Enum (Vertical))	Determines if all of the text is vertical orientation (each line is 90 degrees rotated clockwise, so it goes from top to bottom; each next line is to the left from the previous one).
vert270 (Vertical Text Type Enum (Vertical 270))	Determines if all of the text is vertical orientation (each line is 270 degrees rotated clockwise, so it goes from bottom to top; each next line is to the right from the previous one).
wordArtVert (Vertical Text Type Enum (WordArt Vertical))	Determines if all of the text is vertical ("one letter on top of another").
wordArtVertRtl (Vertical WordArt Right to Left)	Specifies that vertical WordArt should be shown from right to left rather than left to right.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextVerticalType) is located in §A.4.1. *end note*]

20.1.10.83 ST_TextVertOverflowType (Text Vertical Overflow)

This simple type specifies the text vertical overflow.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
clip (Text Overflow Enum (Clip))	Pay attention to top and bottom barriers. Provide no indication that there is text which is not visible.

Enumeration Value	Description
ellipsis (Text Overflow Enum (Ellipsis))	Pay attention to top and bottom barriers. Use an ellipsis to denote that there is text which is not visible.
overflow (Text Overflow Enum (Overflow))	Overflow the text and pay no attention to top and bottom barriers.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextVertOverflowType) is located in §A.4.1. *end note*]

20.1.10.84 ST_TextWrappingType (Text Wrapping Types)

Text Wrapping Types

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
none (Text Wrapping Type Enum (None))	No wrapping occurs on this text body. Words spill out without paying attention to the bounding rectangle boundaries.
square (Text Wrapping Type Enum (Square))	Determines whether we wrap words within the bounding rectangle.

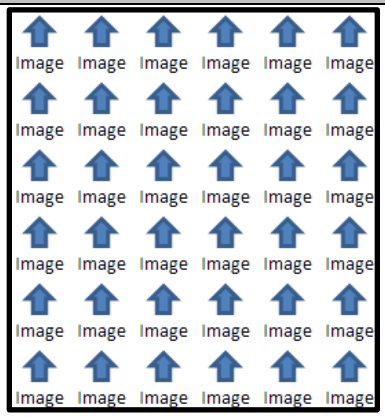
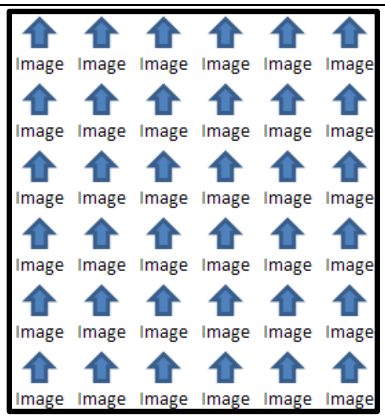
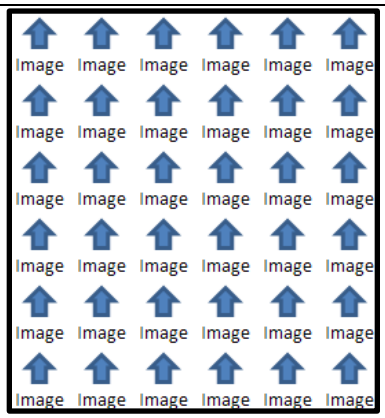
[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextWrappingType) is located in §A.4.1. *end note*]

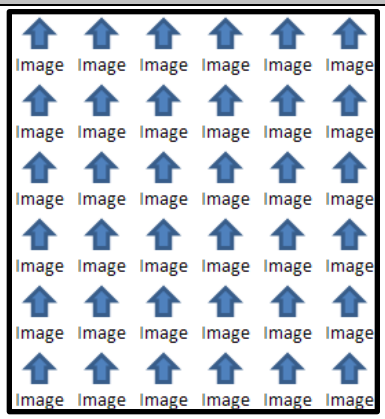
20.1.10.85 ST_TileFlipMode (Tile Flip Mode)

This simple type indicates whether/how to flip the contents of a tile region when using it to fill a larger fill region.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
none (None)	 Tiles are not flipped.
x (Horizontal)	 Tiles are flipped horizontally.
xy (Horizontal and Vertical)	 Tiles are flipped both horizontally and vertically.

Enumeration Value	Description
y (Vertical)	 Tiles are flipped vertically.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TileFlipMode) is located in §A.4.1. *end note*]

20.1.10.86 ST_TextBulletSize (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, with a minimum size of 25% and maximum size of 400%.

This simple type is a union of the following types:

- The ST_TextBulletSizePercent simple type (§20.1.10.62)

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextBulletSize) is located in § A.4.1. *end note*]

20.2 DrawingML - Picture

These elements encompass the definition of pictures within the DrawingML framework. While pictures are in many ways very similar to shapes they have specific properties that are unique in order to optimize for picture-specific scenarios. Some of these properties include Fill behavior, Border behavior and Resize behavior.

20.2.1 Table of Contents

This subclause is informative.

End of informative text.

20.2.2 Elements

The following section defines the Picture portion of the DrawingML framework.

20.2.2.1 blipFill (Picture Fill)

This element specifies the type of picture fill that the picture object has. Because a picture has a picture fill already by default, it is possible to have two fills specified for a picture object. An example of this is shown below.

[*Example*: Consider the picture below that has a blip fill applied to it. The image used to fill this picture object has transparent pixels instead of white pixels.

```
<pic:pic>
  …
  <pic:blipFill>
    <a:blip r:embed="rId2"/>
    <a:stretch>
      <a:fillRect/>
    </a:stretch>
  </pic:blipFill>
  …
</pic:pic>
```

The above picture object is shown as an example of this fill type. *end example*]

[*Example*: Consider now the same picture object but with an additional gradient fill applied within the shape properties portion of the picture.

```
<pic:pic>
  …
  <pic:blipFill>
    <a:blip r:embed="rId2"/>
    <a:stretch>
      <a:fillRect/>
    </a:stretch>
  </pic:blipFill>
  <pic:spPr>
    <a:gradFill>
      <a:gsLst>
        <a:gs pos="0">
          <a:schemeClr val="tx2">
            <a:shade val="50000"/>
          </a:schemeClr>
        </a:gs>
        <a:gs pos="39999">
          <a:schemeClr val="tx2">
            <a:tint val="20000"/>
          </a:schemeClr>
        </a:gs>
        <a:gs pos="70000">
          <a:srgbClr val="C4D6EB"/>
        </a:gs>
        <a:gs pos="100000">
          <a:schemeClr val="bg1"/>
```



```
        </a:gs>
      </a:gsLst>
    </a:gradFill>
  </pic:spPr>
  …
</pic:pic>
```

The above picture object is shown as an example of this double fill type. *end example*]

Attributes	Description
dpi (DPI Setting) Namespace: http://purl.oclc.org/ooxml/drawingml/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: http://purl.oclc.org/ooxml/drawingml/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.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BlipFillProperties) is located in §A.4.1. *end note*]

20.2.2.2 cNvPicPr (Non-Visual Picture Drawing Properties)

This element specifies the non-visual properties for the picture canvas. These properties are to be used by the generating application to determine how certain properties are to be changed for the picture object in question.

[*Example*: Consider the following DrawingML.

```
<pic:pic>
  …
  <pic:nvPicPr>
    <pic:cNvPr id="4" name="Lilly.jpg"/>
    <pic:cNvPicPr>
      <a:picLocks noChangeAspect="1"/>
    </p:cNvPicPr>
    <pic:nvPr/>
  </pic:nvPicPr>
  …
</pic:pic>
```

end example]

Attributes	Description
preferRelativeResize (Relative Resize Preferred) Namespace: http://purl.oclc.org/ooxml/drawingml/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.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualPictureProperties) is located in §A.4.1. *end note*]

20.2.2.3 cNvPr (Non-Visual Drawing Properties)

This element specifies non-visual canvas properties. This allows for additional information that does not affect the appearance of the picture to be stored.

[*Example*: Consider the following DrawingML.

```
<pic:pic>
  …
  <pic:nvPicPr>
    <p:cNvPr id="4" name="Lilly.jpg"/>
  </pic:nvPicPr>
  …
</pic:pic>
```

end example]

Attributes	Description
descr (Alternative Text for Object) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies alternative text for the current DrawingML object, for use by assistive technologies or applications which 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 which 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: http://purl.oclc.org/ooxml/drawingml/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 which 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	Specifies a unique identifier for the current DrawingML object within the current

Attributes	Description
Identifier) Namespace: http://purl.oclc.org/ooxml/drawingml/main	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 (§20.1.10.21).
name (Name) Namespace: http://purl.oclc.org/ooxml/drawingml/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: http://purl.oclc.org/ooxml/drawingml/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.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualDrawingProps) is located in §A.4.1. *end note*]

20.2.2.4 nvPicPr (Non-Visual Picture Properties)

This element specifies the non visual properties for a picture. This allows for additional information that does not affect the appearance of the picture to be stored.

[*Example*: Consider the following DrawingML.

```
<pic:pic>
  …
  <pic:nvPicPr>
    …
  </pic:nvPicPr>
  …
</pic:pic>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_PictureNonVisual) is located in §A.4.2. *end note*]

20.2.2.5 pic (Picture)

This element specifies the existence of a picture object within the document.

[*Example*: Consider the following DrawingML that specifies the existence of a picture within a document. This picture can have non-visual properties, a picture fill as well as shape properties attached to it.

```
<pic:pic>
  <pic:nvPicPr>
    <pic:cNvPr id="4" name="lake.JPG" descr="Picture of a Lake" />
    <pic:cNvPicPr>
      <a:picLocks noChangeAspect="1"/>
    </pic:cNvPicPr>
    <pic:nvPr/>
  </pic:nvPicPr>
  <pic:blipFill>
  …
  </pic:blipFill>
  <pic:spPr>
  …
  </pic:spPr>
</pic:pic>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Picture) is located in §A.4.2. *end note*]

20.2.2.6 spPr (Shape Properties)

This element specifies the visual shape properties that can be applied to a picture. These are the same properties that are allowed to describe the visual properties of a shape but are used here to describe the visual appearance of a picture within a document. This allows for a picture to have both the properties of a shape as well as picture specific properties that are allowed under the pic element.

Attributes	Description
bwMode (Black and White Mode) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeProperties) is located in §A.4.1. *end note*]

20.3 DrawingML - Locked Canvas

Within a DrawingML object, a *locked canvas* allows DrawingML objects to be placed in a format where they can be viewed but not edited by the hosting application. This allows DrawingML objects not supported by an application to be included and viewed in applications where they cannot be edited.

20.3.1 Table of Contents

This subclause is informative.

End of informative text.

20.3.2 Basics

This section specifies a locked canvas within the basic DrawingML framework.

20.3.2.1 lockedCanvas (Locked Canvas Container)

The locked canvas element acts as a container for more advanced drawing objects. The notion of a locked canvas comes from the fact that the generating application opening the file cannot create this object and can

thus not perform edits either. Thus the drawing object is locked from all UI adjustments that would normally take place.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlGroupShape) is located in §A.4.1. *end note*]

20.4 DrawingML - WordprocessingML Drawing

Within a WordprocessingML document, it is possible to include graphical DrawingML objects:

- Pictures (§20.2)
- Locked Canvases (§20.3)
- Diagrams (§21.4)
- Charts (§21.2)

When these objects are present in a word processing document, it is necessary to include information which specifies how the objects shall be positioned relative to the paginated document. [*Example*: Whether the object is displayed in line with text. *end example*]

The WordprocessingML Drawing namespace acts in this capacity, specifying all information necessary to anchor and display DrawingML objects within a word processing document.

[*Example*: Consider a DrawingML picture which must be displayed in the center of the printed page on which it appears, modifying the flow of text as necessary. This object would be specified as follows:

```
<w:r>
  <w:drawing>
    <wp:anchor relativeHeight="10" allowOverlap="true">
      <wp:positionH relativeFrom="margin">
        <wp:align>center</wp:align>
      </wp:positionH>
      <wp:positionV relativeFrom="margin">
        <wp:align>center</wp:align>
      </wp:positionV>
      <wp:extent cx="2441542" cy="1828800"/>
      <wp:wrapSquare wrapText="bothSides"/>
      <a:graphic>
        …
      </a:graphic>
    </wp:anchor>
  </w:drawing>
</w:r>
```

The anchor element (§20.4.2.3) specifies that this object is not positioned in line with text, and its child elements specify that the object is centered on the page horizontally and vertically (§20.4.2.10; §20.4.2.11), and that text can wrap around it in a square (§20.4.2.17). *end example*]

20.4.1 Table of Contents

This subclause is informative.

End of informative text.

20.4.2 Elements

The following elements define the contents of the WordprocessingML Drawing namespace:

20.4.2.1 align (Relative Horizontal Alignment)

This element specifies how a DrawingML object shall be horizontally aligned relative to the horizontal alignment base defined by the parent element. Once an alignment base is defined, this element shall determine how the DrawingML object shall be aligned relative to that location.

[*Example*: Consider a picture in a WordprocessingML document which has been aligned relative to the edge of the page - the left of the page horizontally, and the top of the page vertically. This alignment would be specified as follows:

```
<wp:anchor … >
  <wp:positionH relativeFrom="page">
    <wp:align>left</wp:align>
  </wp:positionH>
  …
</wp:anchor>
```

The align element with a value of `left` specifies that for the horizontal positioning defined by the parent element (in this case, positioning relative to the page), the picture must be aligned to the left edge of the page. *end example*]

The possible values for this element are defined by the ST_AlignH simple type (§20.4.3.1).

[*Note*: The W3C XML Schema definition of this element's content model (ST_AlignH) is located in §A.4.4. *end note*]

20.4.2.2 align (Relative Vertical Alignment)

This element specifies how a DrawingML object shall be vertically aligned relative to the vertical alignment base defined by the parent element. Once an alignment base is defined, this element shall determine how the DrawingML object shall be aligned relative to that location.

[*Example*: Consider a picture in a WordprocessingML document which has been aligned relative to the edge of the page - the left of the page horizontally, and the top of the page vertically. This alignment would be specified as follows:

```
<wp:anchor … >
  <wp:positionV relativeFrom="page">
    <wp:align>top</wp:align>
  </wp:positionH>
  …
</wp:anchor>
```

The align element with a value of top specifies that for the vertical positioning defined by the parent element (in this case, positioning relative to the page), the picture must be aligned to the top edge of the page. *end example*]

The possible values for this element are defined by the ST_AlignV simple type (§20.4.3.2).

[*Note*: The W3C XML Schema definition of this element's content model (ST_AlignV) is located in §A.4.4. *end note*]

20.4.2.3 anchor (Anchor for Floating DrawingML Object)

This element specifies that the DrawingML object located at this position in the document is a floating object. Within a WordprocessingML document, drawing objects can exist in two states:

- *Inline* - The drawing object is in line with the text, and affects the line height and layout of its line (like a character glyph of similar size).
- *Floating* - The drawing object is anchored within the text, but can be absolutely positioned in the document relative to the page.

When this element encapsulates the DrawingML object's information, then all child elements shall dictate the positioning of this object as a floating object on the page.

[*Example*: Consider a WordprocessingML document where the anchor for a floating DrawingML object must be the first piece of run content within a paragraph. That paragraph's content would be specified as follows:


```
<w:p>
  <w:r>
    <w:drawing>
      <wp:anchor … >
        …
      </wp:anchor>
    </w:drawing>
  </w:r>
</w:p>
```

The anchor element, when present as the child element of the drawing element, specifies that this DrawingML object must be positioned as a floating object based on the values of its child elements. *end example*]

Attributes	Description

allowOverlap (Allow Objects to Overlap)	Specifies whether a DrawingML object which intersects another DrawingML object at display time is allowed to overlap the contents of the other DrawingML object. If a DrawingML object cannot overlap other DrawingML object, it shall be repositioned when displayed to prevent this overlap as needed.
	
	[*Example*: Consider a document with two DrawingML objects which are allowed to overlap each other. This would be specified as follows within each object's anchor markup:
	
	`<wp:anchor allowOverlap="true" … >`
	`  …`
	`</wp:anchor>`
	
	The allowOverlap attribute has a value of `true`, which specifies that this object must be allowed to overlap other objects when it is displayed on the document. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

behindDoc (Display Behind Document Text)	Specifies whether this floating DrawingML object is displayed behind the text of the document when the document is displayed. When a DrawingML object is displayed within a WordprocessingML document, that object can intersect with text in the document. This attribute shall determine whether the text or the object is rendered on top in case of overlapping.
	
	[*Example*: Consider a floating DrawingML object which must be displayed above any text which it intersects within the document's content. This setting would be specified as follows:
	
	`<wp:anchor behindDoc="false" … >`
	`  …`
	`</wp:anchor>`

Attributes	Description
	The behindDoc attribute has a value of `false`, which specifies that the DrawingML object is displayed above the text of the document in z-order. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

distB (Distance From Text on Bottom Edge)	Specifies the minimum distance which shall be maintained between the bottom edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents.
	
	The distance shall be measured in EMUs (English Metric Units).
	
	If this object is an inline object (i.e. has a parent element of inline), then this value shall not have any effect when displaying the object in line with text, but can be maintained and used if the object is subsequently changed to floating. If the wrapping element [*Example*: wrapThrough or wrapSquare *end example*] present as a child element also has a distance from text, then this value shall be ignored.
	
	[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its bottom edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor distB="457200" … >`
	`  …`
	`</wp:anchor>`
	
	The distB attribute specifies that the padding distance must be `457200` EMUs or one-half of an inch. *end example*]
	
	The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

distL (Distance From Text on Left Edge)	Specifies the minimum distance which shall be maintained between the left edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents. The distance shall be measured in EMUs (English Metric Units). If this object is an inline object (i.e. has a parent element of inline), then this value shall not have any effect when displaying the object in line with text, but can be maintained and used if the object is subsequently changed to floating. If the wrapping element [*Example*: wrapThrough or wrapSquare *end example*] present as a child element also has a distance from text, then this value shall be ignored. [*Example*: Consider a floating DrawingML object which must have one-quarter of an inch of padding between its left edge and the nearest text. This setting would be specified as follows:

Attributes	Description
	<wp:anchor distL="228600" … > … </wp:anchor> The distL attribute specifies that the padding distance must be 228600 EMUs or one-quarter of an inch. *end example*] The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).
distR (Distance From Text on Right Edge)	Specifies the minimum distance which shall be maintained between the right edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents. The distance shall be measured in EMUs (English Metric Units). If this object is an inline object (i.e. has a parent element of inline), then this value shall not have any effect when displaying the object in line with text, but can be maintained and used if the object is subsequently changed to floating. If the wrapping element [*Example*: wrapThrough or wrapSquare *end example*] present as a child element also has a distance from text, then this value shall be ignored. [*Example*: Consider a floating DrawingML object which must have one-quarter of an inch of padding between its right edge and the nearest text. This setting would be specified as follows: <wp:anchor distR="228600" … > … </wp:anchor> The distR attribute specifies that the padding distance must be 228600 EMUs or one-quarter of an inch. *end example*] The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).
distT (Distance From Text on Top Edge)	Specifies the minimum distance which shall be maintained between the top edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents. The distance shall be measured in EMUs (English Metric Units). If this object is an inline object (i.e. has a parent element of inline), then this value shall not have any effect when displaying the object in line with text, but can be maintained and used if the object is subsequently changed to floating. If the wrapping element [*Example*: wrapThrough or wrapSquare *end example*] present as a child element also has a distance from text, then this value shall be ignored.

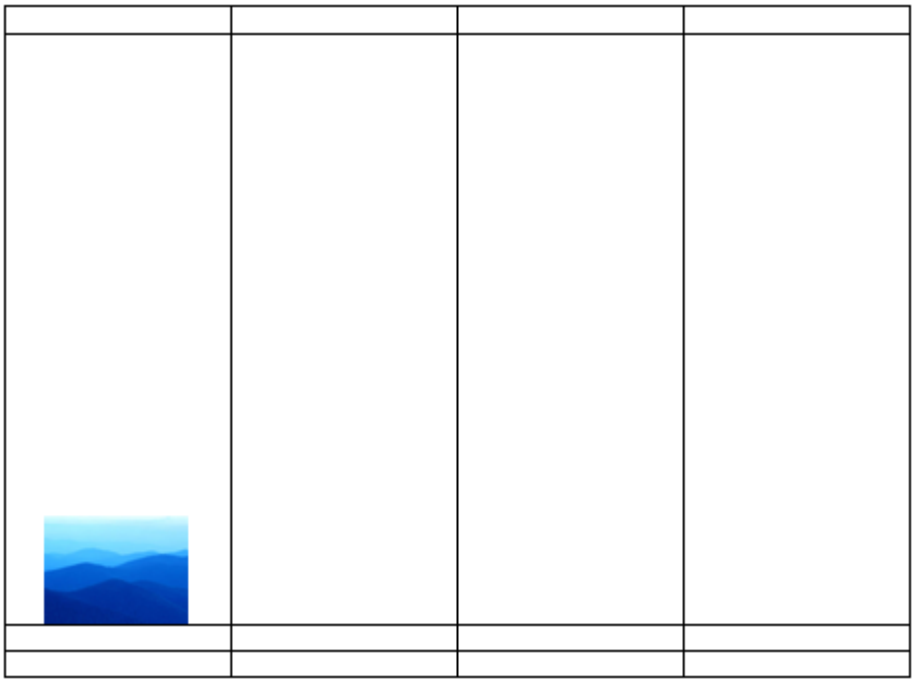
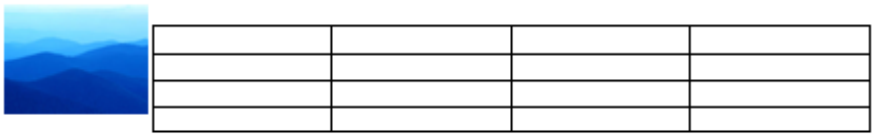
Attributes	Description

	[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its top edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor distT="457200" … >`
	`  …`
	`</wp:anchor>`
	
	The distT attribute specifies that the padding distance must be 457200 EMUs or one-half of an inch. *end example*]
	
	The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

hidden (Hidden)	Specifies whether this floating DrawingML object is displayed. When a DrawingML object is displayed within a WordprocessingML document, that object can be hidden (i.e. present, but not visible). This attribute shall determine 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 a floating DrawingML object which must be hidden within the document's content. This setting would be specified as follows:
	
	`<wp:anchor hidden="true" … >`
	`  …`
	`</wp:anchor>`
	
	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.

layoutInCell (Layout In Table Cell)	Specifies how this DrawingML object behaves when its anchor is located in a table cell; and its specified position would cause it to intersect with a table cell displayed in the document. That behavior shall be as follows: • When this attribute has a value of true, then the object shall be positioned within the existing table cell, causing the cell to be resized as needed. This means that all positioning shall be relative to the cell and not the line on which the table appears. • When this attribute has a value of false, then the object shall be positioned as specified, but the table shall be resized and/or relocated within the document as needed to accommodate the object. This means that all positioning shall be

Attributes	Description

	relative to the line on which the table appears and not the cell in which the anchor is present.
	
	[*Example*: Consider a DrawingML picture which must be displayed in the center of the document. If the object is contained within a table and is defined as follows:
	
	`<wp:anchor layoutInCell="true" … >`
	`…`
	`</wp:anchor>`
	
	The layoutInCell attribute has a value of `true`, which specifies that the object can be placed within the cell if needed, for example:
	
	 
	
	If the layoutInCell attribute was now set to `false`, the object must be laid out outside of the cell, causing the table to be repositioned:
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

locked (Lock	Specifies that the anchor location for this object shall not be modified at runtime when

Attributes	Description

Anchor)	an application edits the contents of this document. [*Guidance*: An application might have automatic behaviors which reposition the anchor for a DrawingML 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*]
	
	[*Example*: Consider a floating DrawingML object which must have its anchor locked at the current location. This setting would be specified as follows:
	
	`<wp:anchor locked="true" … >`
	`  …`
	`</wp:anchor>`
	
	The locked attribute has a value of `true`, which specifies that the DrawingML object's current anchor location must not be changed by applications editing this content. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

relativeHeight (Relative Z-Ordering Position)	Specifies the relative Z-ordering of all DrawingML objects in this document. Each floating DrawingML object shall have a Z-ordering value, which determines which object is displayed when any two objects intersect. Higher values shall indicate higher Z-order; lower values shall indicate lower Z-order.
	
	This attribute shall only indicate the Z-order with respect to other objects in the document which have an identical behindDoc attribute value. All objects with a behindDoc value of `false` shall be displayed above elements with a value of `true`.
	
	[*Example*: Consider two floating DrawingML objects as follows:
	
	`<wp:anchor relativeHeight="5" … >`
	`  …`
	`</wp:anchor>`
	
	`…`
	
	`<wp:anchor relativeHeight="8" … >`
	`  …`
	`</wp:anchor>`
	
	The relativeHeight attribute of the second object is 8, which specifies that the second DrawingML object must be at a higher Z-order than the first and must be displayed whenever the two overlap. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

simplePos (Page	Specifies that this object shall be positioned using the positioning information in the

Attributes	Description

Positioning)	simplePos child element (§20.4.2.13). This positioning, when specified, positions the object on the page by placing its top left point at the x-y coordinates specified by that element.
	
	If this element is omitted, then this object shall not use the simple positioning information in the simplePos element, even when present.
	[*Example*: Consider a floating DrawingML object which must be positioned at the top left corner of the page using simple positioning. This setting would be specified as follows:
	
	`<wp:anchor simplePos="true" … >`
	`  <wp:simplePos x="0" y="0" />`
	`  …`
	`</wp:anchor>`
	
	The simplePos attribute has a value of `true`, which specifies that the DrawingML object's current position must be dictated by the simplePos element, and hence placed at 0,0. *end example*]
	
	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_Anchor) is located in §A.4.4. *end note*]

20.4.2.4 cNvGraphicFramePr (Common DrawingML Non-Visual Properties)

This element specifies common non-visual DrawingML object properties for the parent DrawingML object. These properties are specified as child elements of this element.

[*Example*: Consider a DrawingML object in a WordprocessingML document defined as follows:

```
<wp:inline>
  …
  <wp:cNvGraphicFramePr>
    <a:graphicFrameLocks … />
  </wp:cNvGraphicFramePr>
</wp:inline>
```

The cNvGraphicFramePr element contains a set of common non-visual properties as defined by DrawingML. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualGraphicFrameProperties) is located in §A.4.1. *end note*]

20.4.2.5 docPr (Drawing Object Non-Visual Properties)

This element specifies non-visual object properties for the parent DrawingML object. These properties are specified as child elements of this element.

[*Example*: Consider a DrawingML object in a WordprocessingML document defined as follows:

```
<wp:inline>
  …
  <wp:docPr id="1" name="Example Object">
    <a:hlinkClick … />
    <a:hlinkHover … />
  </wp:docPr>
</wp:inline>
```

The docPr element contains a set of common non-visual properties for this object. *end example*]

Attributes	Description
descr (Alternative Text for Object) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies alternative text for the current DrawingML object, for use by assistive technologies or applications which 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 which 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: http://purl.oclc.org/ooxml/drawingml/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 which 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*]

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Unique Identifier) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.21).
name (Name) Namespace: http://purl.oclc.org/ooxml/drawingml/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: http://purl.oclc.org/ooxml/drawingml/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.

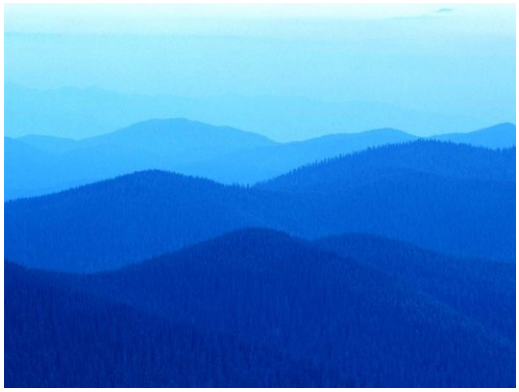
[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualDrawingProps) is located in §A.4.1. *end note*]

20.4.2.6 effectExtent (Object Extents Including Effects)

This element specifies the additional extent which shall be added to each edge of the image (top, bottom, left, right) in order to compensate for any drawing effects applied to the DrawingML object.

The extent element (§20.4.2.7) specifies the size of the actual DrawingML object; however, an object can have effects applied which change its overall size [*Example*: A reflection and/or shadow effect. *end example*]. The additional size for each edge of the shape shall be stored on this element, and used to calculate the appropriate wrapping for wrap types without a wrapping polygon and the appropriate line height for inline objects.

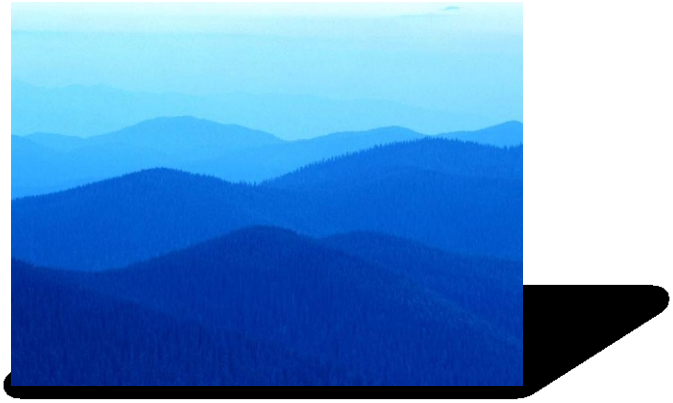
[*Example*: Consider the following DrawingML image:



This object has no effects, and hence would have the following effect extents:

```
<wp:effectExtents b="0" t="0" l="0" r="0" />
```

However, if a shadow effect was applied which added effects to the right of the image:



Then the additional extent the right side would be specified in the r attribute on this element:

```
<wp:effectExtents b="0" t="0" l="0" r="695325" />
```


The r attribute has a value of 695325, specifying that that 695325 EMUs must be added to the right side of the image. *end example*]

Attributes	Description
b (Additional Extent on Bottom Edge)	Specifies the additional length, in EMUs, which shall be added to the bottom edge of the DrawingML object to determine its actual bottom edge including effects. [*Example*: Consider the following DrawingML image:  This image has an effect on all four sides, resulting in the following markup: `<wp:effectExtent l="504825" t="447675" r="771525" b="809625" />` The b attribute value of 809625 specifies that 809625 additional EMUs must be added to the bottom of the image to compensate for the effects on the image. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
l (Additional Extent on Left Edge)	Specifies the additional length, in EMUs, which shall be added to the bottom edge of the DrawingML object to determine its actual bottom edge including effects. [*Example*: Consider the following DrawingML image:

Attributes	Description
	 This image has an effect on all four sides, resulting in the following markup: `<wp:effectExtent l="504825" t="447675" r="771525" b="809625" />` The l attribute value of 504825 specifies that 504825 additional EMUs must be added to the bottom of the image to compensate for the effects on the image. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
r (Additional Extent on Right Edge)	Specifies the additional length, in EMUs, which shall be added to the bottom edge of the DrawingML object to determine its actual bottom edge including effects. [*Example*: Consider the following DrawingML image:  This image has an effect on all four sides, resulting in the following markup:

Attributes	Description
	<wp:effectExtent l="504825" t="447675" r="771525" b="809625" /> The r attribute value of 771525 specifies that 771525 additional EMUs must be added to the bottom of the image to compensate for the effects on the image. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
t (Additional Extent on Top Edge)	Specifies the additional length, in EMUs, which shall be added to the bottom edge of the DrawingML object to determine its actual bottom edge including effects. [*Example*: Consider the following DrawingML image:  This image has an effect on all four sides, resulting in the following markup: <wp:effectExtent l="504825" t="447675" r="771525" b="809625" /> The t attribute value of 447675 specifies that 447675 additional EMUs must be added to the bottom of the image to compensate for the effects on the image. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_EffectExtent) is located in §A.4.4. *end note*]

20.4.2.7 extent (Drawing Object Size)

This element specifies the extents of the parent DrawingML object within the document (i.e. its final height and width).

[*Example*: Consider a DrawingML picture which is present in a WordprocessingML document and has an equal height and width. This object would be specified as follows:

```
<wp:anchor relativeHeight="10" allowOverlap="true">
  …
  <wp:extent cx="1828800" cy="1828800"/>
  …
</wp:anchor>
```

The extent element specifies via its attributes that this object has a height and width of 1828800 EMUs (English Metric Units). *end example*]

Attributes	Description
cx (Extent Length) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.41).
cy (Extent Width) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositiveSize2D) is located in §A.4.1. *end note*]

20.4.2.8 inline (Inline DrawingML Object)

This element specifies that the DrawingML object located at this position in the document is an inline object. Within a WordprocessingML document, drawing objects can exist in two states:

- *Inline* - The drawing object is in line with the text, and affects the line height and layout of its line (like a character glyph of similar size).
- *Floating* - The drawing object is anchored within the text, but can be absolutely positioned in the document relative to the page.

When this element encapsulates the DrawingML object's information, then all child elements shall dictate the positioning of this object in line with text.

[*Example*: Consider a WordprocessingML document where an inline DrawingML object must be the first piece of run content within a paragraph. That paragraph's content would be specified as follows:

```
<w:p>
  <w:r>
    <w:drawing>
      <wp:inline>
        …
      </wp:inline>
    </w:drawing>
  </w:r>
</w:p>
```

The inline element, when present as the child element of the drawing element, specifies that this DrawingML object must be positioned in line with the text of this paragraph, modifying line heights, etc. as necessary. *end example*]

Attributes	Description

distB (Distance From Text on Bottom Edge)	Specifies the minimum distance which shall be maintained between the bottom edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents.
	
	The distance shall be measured in EMUs (English Metric Units).
	
	If this object is an inline object (i.e. has a parent element of inline), then this value shall not have any effect when displaying the object in line with text, but can be maintained and used if the object is subsequently changed to floating. If the wrapping element [*Example*: wrapThrough or wrapSquare *end example*] present as a child element also has a distance from text, then this value shall be ignored.
	
	[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its bottom edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor distB="457200" … >`
	`  …`
	`</wp:anchor>`

Attributes	Description
	The distB attribute specifies that the padding distance must be 457200 EMUs or one-half of an inch. *end example*] The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

distL (Distance From Text on Left Edge)	Specifies the minimum distance which shall be maintained between the left edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents.
	
	The distance shall be measured in EMUs (English Metric Units).
	
	If this object is an inline object (i.e. has a parent element of inline), then this value shall not have any effect when displaying the object in line with text, but can be maintained and used if the object is subsequently changed to floating. If the wrapping element [*Example*: wrapThrough or wrapSquare *end example*] present as a child element also has a distance from text, then this value shall be ignored.
	
	[*Example*: Consider a floating DrawingML object which must have one-quarter of an inch of padding between its left edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor distL="228600" … >`
	`  …`
	`</wp:anchor>`
	
	The distL attribute specifies that the padding distance must be 228600 EMUs or one-quarter of an inch. *end example*]
	
	The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

distR (Distance From Text on Right Edge)	Specifies the minimum distance which shall be maintained between the right edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents. The distance shall be measured in EMUs (English Metric Units). If this object is an inline object (i.e. has a parent element of inline), then this value shall not have any effect when displaying the object in line with text, but can be maintained and used if the object is subsequently changed to floating. If the wrapping element [*Example*: wrapThrough or wrapSquare *end example*] present as a child element also has a distance from text, then this value shall be ignored. [*Example*: Consider a floating DrawingML object which must have one-quarter of an inch of padding between its right edge and the nearest text. This setting would be specified as follows:

Attributes	Description
	<wp:anchor distR="228600" … > … </wp:anchor> The distR attribute specifies that the padding distance must be 228600 EMUs or one-quarter of an inch. *end example*] The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).
distT (Distance From Text on Top Edge)	Specifies the minimum distance which shall be maintained between the top edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents. The distance shall be measured in EMUs (English Metric Units). If this object is an inline object (i.e. has a parent element of inline), then this value shall not have any effect when displaying the object in line with text, but can be maintained and used if the object is subsequently changed to floating. If the wrapping element [*Example*: wrapThrough or wrapSquare *end example*] present as a child element also has a distance from text, then this value shall be ignored. [*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its top edge and the nearest text. This setting would be specified as follows: <wp:anchor distT="457200" … > … </wp:anchor> The distT attribute specifies that the padding distance must be 457200 EMUs or one-half of an inch. *end example*] The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Inline) is located in §A.4.4. *end note*]

20.4.2.9 lineTo (Wrapping Polygon Line End Position)

This element specifies a single point on the wrapping polygon for a DrawingML object. This point shall be the termination of the edge of the wrapping polygon started by the previous start or lineTo element in document order, and shall be the origin of the next edge on the same polygon.

The attributes on this element shall dictate the position of the point relative to the upper-left corner of the actual object.

[*Example*: Consider the following basic wrapping polygon for a DrawingML object:

```
<wp:wrapPolygon>
  <wp:start x="0" y="0" />
  <wp:lineTo x="0" y="100" />
  <wp:lineTo x="100" y="100" />
  <wp:lineTo x="100" y="0" />
  <wp:lineTo x="0" y="0" />
</wp:wrapPolygon>
```

The lineTo element defines each point of the wrapping polygon (in this case, the four points of the wrapping square). *end example*]

Attributes	Description
x (X-Axis Coordinate) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.16).
y (Y-Axis Coordinate) Namespace: http://purl.oclc.org/ooxml/drawingml/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 y attribute defines a y-coordinate of `100`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Point2D) is located in §A.4.1. *end note*]

20.4.2.10 positionH (Horizontal Positioning)

This element specifies the horizontal positioning of a floating DrawingML object within a WordprocessingML document. This positioning is specified in two parts:

- Positioning Base - The relativeFrom attribute on this element specifies the part of the document from which the positioning shall be calculated.
- Positioning - The child element of this element (align or posOffset) specifies how the object is positioned relative to that base.

[*Example*: Consider a DrawingML picture which must be displayed in the center of the printed page on which it appears, modifying the flow of text as necessary. This object would be specified as follows:

```
<wp:anchor … >
  <wp:positionH relativeFrom="margin">
    <wp:align>center</wp:align>
  </wp:positionH>
  <wp:positionV relativeFrom="margin">
    <wp:align>center</wp:align>
  </wp:positionV>
</wp:anchor>
```

The positionH element specifies that the object is horizontally positioned relative to the margin via the relativeFrom attribute; and that the alignment relative to the margin is centered via the align element. *end example*]

Attributes	Description

relativeFrom (Horizontal Position Relative Base)	Specifies the base to which the relative horizontal positioning of this object shall be calculated.
	
	[*Example*: Consider a DrawingML picture which must be displayed at the bottom center of the page. This object would be specified as follows:
	
	`<wp:anchor … >`
	`  <wp:positionH relativeFrom="page">`
	`  <wp:align>center</wp:align>`
	`  </wp:positionH>`
	`  …`
	`</wp:anchor>`
	
	The relativeFrom attribute specifies that the object is horizontally positioned relative to the page. *end example*]
	
	The possible values for this attribute are defined by the ST_RelFromH simple type (§20.4.3.4).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PosH) is located in §A.4.4. *end note*]

20.4.2.11 positionV (Vertical Positioning)

This element specifies the vertical positioning of a floating DrawingML object within a WordprocessingML document. This positioning is specified in two parts:

- Positioning Base - The relativeFrom attribute on this element specifies the part of the document from which the positioning shall be calculated.
- Positioning - The child element of this element (align or posOffset) specifies how the object is positioned relative to that base.

[*Example*: Consider a DrawingML picture which must be displayed in the center of the printed page on which it appears, modifying the flow of text as necessary. This object would be specified as follows:

```
<wp:anchor … >
  <wp:positionH relativeFrom="margin">
    <wp:align>center</wp:align>
  </wp:positionH>
  <wp:positionV relativeFrom="margin">
    <wp:align>center</wp:align>
  </wp:positionV>
</wp:anchor>
```

The positionV element specifies that the object is vertically positioned relative to the margin via the relativeFrom attribute; and that the alignment relative to the margin is centered via the align element. *end example*]

Attributes	Description

relativeFrom (Vertical Position Relative Base)	Specifies the base to which the relative vertical positioning of this object shall be calculated.
	
	[*Example*: Consider a DrawingML picture which must be displayed at the bottom center of the page margins. This object would be specified as follows:
	
	`<wp:anchor … >`
	`…`
	`<wp:positionV relativeFrom="margin">`
	`<wp:align>bottom</wp:align>`
	`</wp:positionV>`
	`</wp:anchor>`
	
	The relativeFrom attribute specifies that the object is horizontally positioned relative to the margin. *end example*]

Attributes	Description
	The possible values for this attribute are defined by the ST_RelFromV simple type (§20.4.3.5).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PosV) is located in §A.4.4. *end note*]

20.4.2.12 posOffset (Absolute Position Offset)

This element specifies an absolute measurement for the positioning of a floating DrawingML object within a WordprocessingML document. This measurement shall be calculated relative to the top left edge of the positioning base specified by the parent element's relativeFrom attribute.

[*Example*: Consider a DrawingML picture which must be displayed one inch from the top of the page, and one-half of an inch from the left edge of the page. This object would be specified as follows:

```
<wp:anchor … >
  <wp:positionH relativeFrom="page">
    <wp:posOffset>914400</wp:posOffset>
  </wp:positionH>
  <wp:positionV relativeFrom="page">
    <wp:posOffset>457200</wp:posOffset>
  </wp:positionV>
</wp:anchor>
```

The posOffset element specifies the absolute positioning of the object relative to the top-left edge of the page in EMUs. *end example*]

The possible values for this element are defined by the ST_PositionOffset simple type (§20.4.3.3).

[*Note*: The W3C XML Schema definition of this element's content model (ST_PositionOffset) is located in §A.4.4. *end note*]

20.4.2.13 simplePos (Simple Positioning Coordinates)

This element specifies the coordinates at which a DrawingML object shall be positioned relative to the top-left edge of its page, when the simplePos attribute is specified on the anchor element (§20.4.2.3).

[*Example*: Consider a floating DrawingML object which must be positioned at the top left corner of the page using simple positioning. This setting would be specified as follows:

```
<wp:anchor simplePos="true" … >
  <wp:simplePos x="0" y="0" />
  …
</wp:anchor>
```

The simplePos attribute has a value of `true`, which specifies that the DrawingML object's current position must be dictated by the simplePos element, and hence placed at `0,0`. *end example*]

Attributes	Description
x (X-Axis Coordinate) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.16).
y (Y-Axis Coordinate) Namespace: http://purl.oclc.org/ooxml/drawingml/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 y attribute defines a y-coordinate of `100`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Point2D) is located in §A.4.1. *end note*]

20.4.2.14 start (Wrapping Polygon Start)

This element specifies the starting point on the wrapping polygon for a DrawingML object. This point shall be the start and termination of the wrapping polygon for the parent object.

The attributes on this element shall dictate the position of the point relative to the upper-left corner of the actual object.

[*Example*: Consider the following basic wrapping polygon for a DrawingML object:

```
<wp:wrapPolygon>
  <wp:start x="0" y="0" />
  <wp:lineTo x="0" y="100" />
  <wp:lineTo x="100" y="100" />
  <wp:lineTo x="100" y="0" />
  <wp:lineTo x="0" y="0" />
</wp:wrapPolygon>
```

The start element defines the start and end of the wrapping polygon (in this case, the four points of the wrapping square). *end example*]

Attributes	Description
x (X-Axis Coordinate) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.16).
y (Y-Axis Coordinate) Namespace: http://purl.oclc.org/ooxml/drawingml/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 y attribute defines a y-coordinate of `100`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Point2D) is located in §A.4.1. *end note*]

20.4.2.15 wrapNone (No Text Wrapping)

This element specifies that the parent DrawingML object shall not cause any text wrapping within the contents of the host WordprocessingML document based on its display location. In effect, this setting shall place the object in one of two locations:

- If the behindDoc attribute on the parent element is `true`, then the object shall be positioned behind the text as it is normally displayed.
- If the behindDoc attribute on the parent element is `false`, then the object shall be positioned in front of the text as it is normally displayed.

[*Example*: Consider a DrawingML picture which must be displayed in front of any text on the page. This object would be specified as follows:

```
<wp:anchor relativeHeight="10" behindDoc="false">
  …
  <wp:wrapNone/>
</wp:anchor>
```

The wrapNone element specifies that the DrawingML object must not cause any text wrapping, and since the behindDoc attribute is `false`, the object must be displayed in front of the text of the document. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_WrapNone) is located in §A.4.4. *end note*]

20.4.2.16 wrapPolygon (Wrapping Polygon)

This element specifies the wrapping polygon which shall be used to determine the extents to which text can wrap around the specified object in the document. This polygon shall be defined by the following:

- The start element defines the coordinates of the origin of the wrap polygon
- Two or more lineTo elements define the point of the wrap polygon

If the set of child elements does not result in a closed polygon (the last lineTo element does not return to the position specified by the start element), then a single additional line shall be inferred as needed to close the wrapping polygon.

[*Example*: Consider the following basic wrapping polygon for a DrawingML object:

```
<wp:wrapPolygon>
  <wp:start x="0" y="0" />
  <wp:lineTo x="0" y="100" />
  <wp:lineTo x="100" y="100" />
  <wp:lineTo x="100" y="0" />
  <wp:lineTo x="0" y="0" />
</wp:wrapPolygon>
```

The wrapPolygon element defines the object's text wrapping polygon (in this case, the four points of a square). *end example*]

Attributes	Description
edited (Wrapping	Specifies that the wrap points for the wrapping polygon have been edited, and the

Attributes	Description

Points Modified)	resulting extents shall be recalculated to compensate when the document is next opened.
	
	[*Example*: Consider the following basic wrapping polygon for a DrawingML object:
	
	`<wp:wrapPolygon edited="true">`
	`<wp:start x="0" y="0" />`
	`<wp:lineTo x="0" y="100" />`
	`<wp:lineTo x="50" y="50" />`
	`<wp:lineTo x="0" y="0" />`
	`</wp:wrapPolygon>`
	
	The edited attribute specifies that these wrap points have been changed since the document was last rendered. *end example*]
	
	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_WrapPath) is located in §A.4.4. *end note*]

20.4.2.17 wrapSquare (Square Wrapping)

This element specifies that text shall wrap around a virtual rectangle bounding this object. The bounds of the wrapping rectangle shall be dictated by the extents including the addition of the effectExtent element as a child of this element (if present) or the effectExtent present on the parent element.

[*Example*: Consider a DrawingML object using square wrapping and defined as follows:

```
<wp:anchor … >
  …
  <wp:wrapSquare wrapText="bothSides" />
</wp:anchor>
```

The wrapSquare element specifies that text must wrap around both sides of a rectangle around this object which includes its effect extents. *end example*]

Attributes	Description
distB (Distance From Text on Bottom Edge)	Specifies the minimum distance which shall be maintained between the bottom edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents. The distance shall be measured in EMUs (English Metric Units). [*Example*: Consider a floating DrawingML object which must have one-half of an inch of

Attributes	Description

	padding between its bottom edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor … >`
	`  …`
	`  <wp:wrapSquare distB="457200" … />`
	`</wp:anchor>`
	
	The distB attribute specifies that the padding distance must be 457200 EMUs or one-half of an inch. *end example*]
	
	The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

distL (Distance From Text on Left Edge)	Specifies the minimum distance which shall be maintained between the left edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents.
	
	The distance shall be measured in EMUs (English Metric Units).
	
	[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its left edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor … >`
	`  …`
	`  <wp:wrapSquare distL="457200" … />`
	`</wp:anchor>`
	
	The distL attribute specifies that the padding distance must be 457200 EMUs or one-half of an inch. *end example*]
	
	The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

distR (Distance From Text on Right Edge)	Specifies the minimum distance which shall be maintained between the right edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents.
	
	The distance shall be measured in EMUs (English Metric Units).
	
	[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its right edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor … >`
	`  …`
	`  <wp:wrapSquare distR="457200" … />`

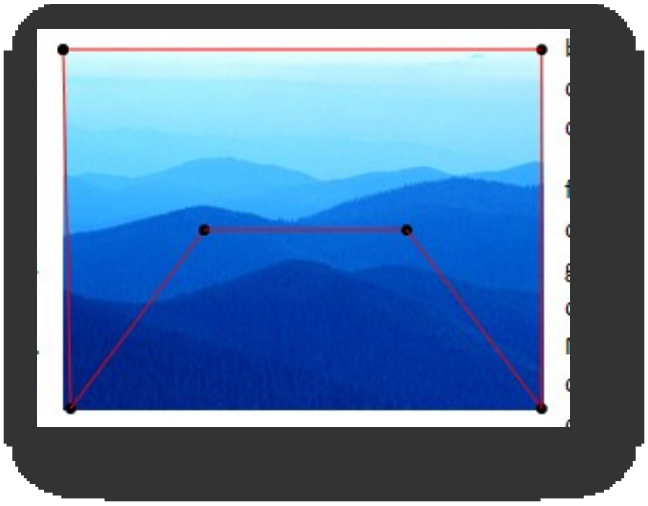
Attributes	Description
	</wp:anchor> The distR attribute specifies that the padding distance must be 457200 EMUs or one-half of an inch. *end example*] The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).
distT (Distance From Text (Top))	Specifies the minimum distance which shall be maintained between the top edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents. The distance shall be measured in EMUs (English Metric Units). [*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its top edge and the nearest text. This setting would be specified as follows: <wp:anchor … > … <wp:wrapSquare distT="457200" … /> </wp:anchor> The distT attribute specifies that the padding distance must be 457200 EMUs or one-half of an inch. *end example*] The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).
wrapText (Text Wrapping Location)	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 would be specified as follows: <wp:anchor … > … <wp:wrapSquare wrapText="left" … /> </wp:anchor> The wrapText attribute value of left specifies that text must only wrap around the let side of the object. *end example*] The possible values for this attribute are defined by the ST_WrapText simple type (§20.4.3.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_WrapSquare) is located in §A.4.4. *end note*]

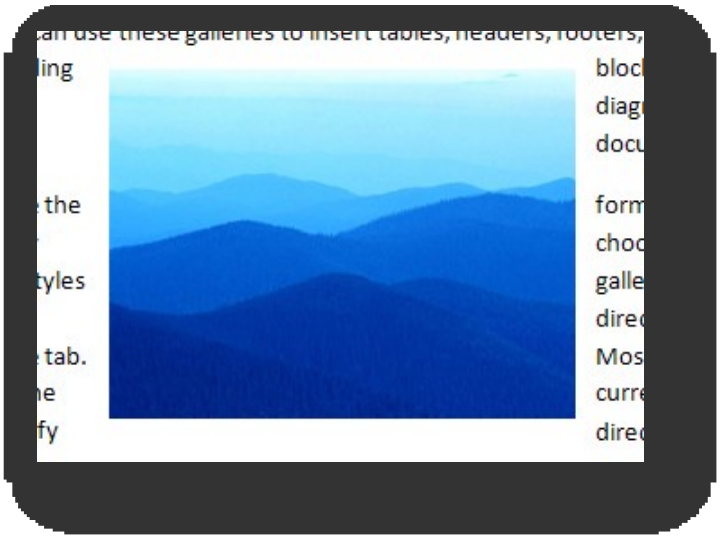
20.4.2.18 wrapThrough (Through Wrapping)

This element specifies that text shall wrap around the wrapping polygon bounding this object as defined by the child wrapPolygon element. When this element specifies a wrapping polygon, it shall allow text to wrap within the object's maximum left and right extents.

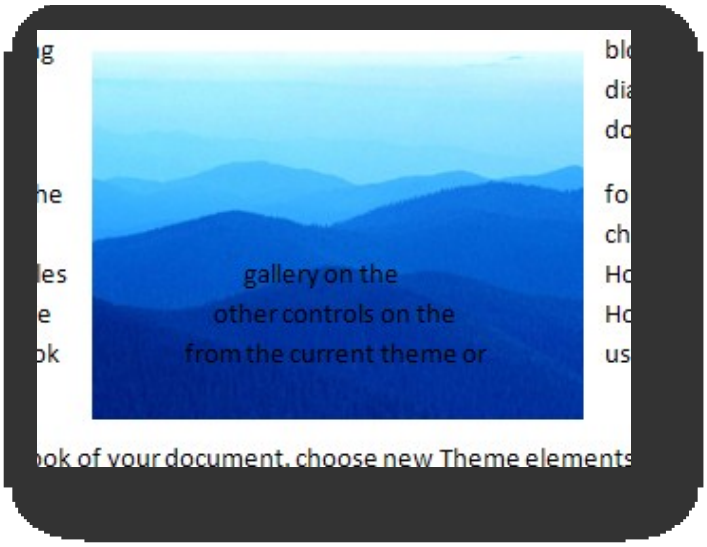
[*Example*: Consider an object with the following wrap points:



If this object uses tight wrapping, then text cannot be placed within the maximum left and right extents of the wrap polygon at any location:



However, with through wrapping:



end example]

[*Example*: Consider a DrawingML object using through wrapping and defined as follows:

```
<wp:anchor … >
  …
  <wp:wrapThrough wrapText="bothSides">
    …
  </wp:wrapThrough>
</wp:anchor>
```

The wrapThrough element specifies that text must wrap through this object as defined by its wrap polygon. *end example*]

Attributes	Description

distL (Distance From Text on Left Edge)	Specifies the minimum distance which shall be maintained between the left edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents.
	
	The distance shall be measured in EMUs (English Metric Units).
	
	[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its left edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor … >`
	`  …`
	`  <wp:wrapThrough distL="457200" … />`
	`</wp:anchor>`
	
	The distL attribute specifies that the padding distance must be 457200 EMUs or one-half

Attributes	Description
	of an inch. *end example*] The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

distR (Distance From Text on Right Edge)	Specifies the minimum distance which shall be maintained between the right edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents.
	
	The distance shall be measured in EMUs (English Metric Units).
	
	[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its right edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor … >`
	`  …`
	`  <wp:wrapThrough distR="457200" … />`
	`</wp:anchor>`
	
	The distR attribute specifies that the padding distance must be `457200` EMUs or one-half of an inch. *end example*]
	
	The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

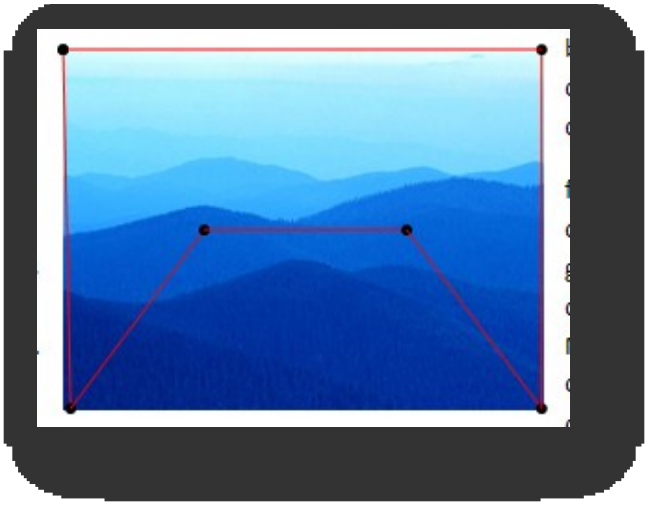
wrapText (Text Wrapping Location)	Specifies how text shall wrap around the object's left and right sides.
	
	[*Example*: Consider a floating DrawingML object which must allow text to wrap around its left side only. This setting would be specified as follows:
	
	`<wp:anchor … >`
	`  …`
	`  <wp:wrapThrough wrapText="left" … />`
	`</wp:anchor>`
	
	The wrapText attribute value of `left` specifies that text must only wrap around the let side of the object. *end example*]
	
	The possible values for this attribute are defined by the ST_WrapText simple type (§20.4.3.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_WrapThrough) is located in §A.4.4. *end note*]

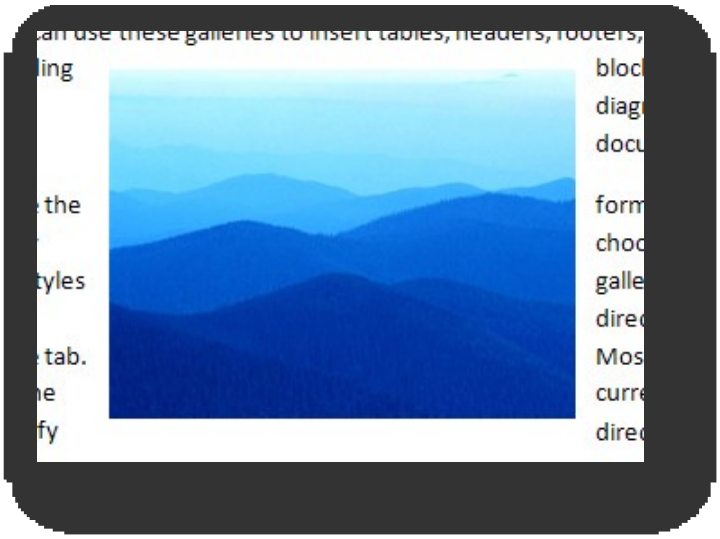
20.4.2.19 wrapTight (Tight Wrapping)

This element specifies that text shall wrap around the wrapping polygon bounding this object as defined by the child wrapPolygon element. When this element specifies a wrapping polygon, it shall not allow text to wrap within the object's maximum left and right extents.

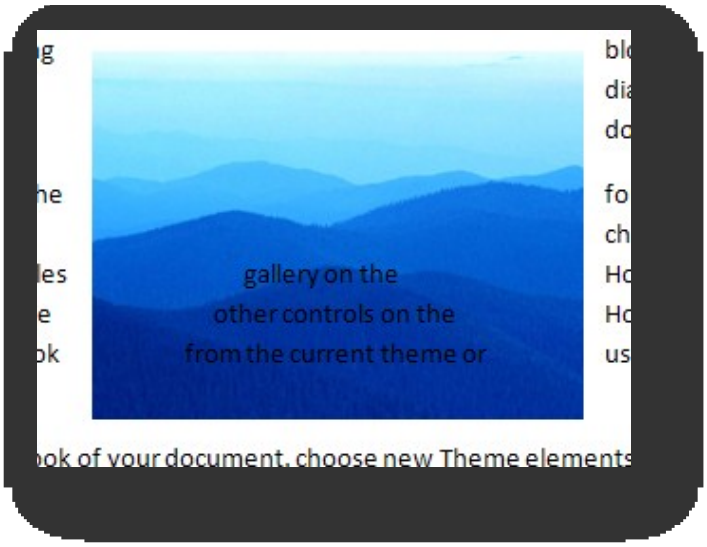
[*Example*: Consider an object with the following wrap points:



If this object uses tight wrapping, then text cannot be placed within the maximum left and right extents of the wrap polygon at any location:



However, with through wrapping:



end example]

[*Example*: Consider a DrawingML object using tight wrapping and defined as follows:

```
<wp:anchor … >
  …
  <wp:wrapTight wrapText="bothSides">
    …
  </wp:wrapTight>
</wp:anchor>
```

The wrapTight element specifies that text must wrap through this object as defined by its wrap polygon. *end example*]

Attributes	Description

distL (Distance From Test on Left Edge)	Specifies the minimum distance which shall be maintained between the left edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents.
	
	The distance shall be measured in EMUs (English Metric Units).
	
	[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its left edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor … >`
	`  …`
	`  <wp:wrapTight distL="457200" … />`
	`</wp:anchor>`
	
	The distL attribute specifies that the padding distance must be 457200 EMUs or one-half

Attributes	Description
	of an inch. *end example*] The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

distR (Distance From Text on Right Edge)	Specifies the minimum distance which shall be maintained between the right edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents.
	
	The distance shall be measured in EMUs (English Metric Units).
	
	[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its right edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor … >`
	`  …`
	`  <wp:wrapTight distR="457200" … />`
	`</wp:anchor>`
	
	The distR attribute specifies that the padding distance must be `457200` EMUs or one-half of an inch. *end example*]
	
	The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

wrapText (Text Wrapping Location)	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 would be specified as follows:
	
	`<wp:anchor … >`
	`  …`
	`  <wp:wrapTight wrapText="left" … />`
	`</wp:anchor>`
	
	The wrapText attribute value of `left` specifies that text must only wrap around the let side of the object. *end example*]
	
	The possible values for this attribute are defined by the ST_WrapText simple type (§20.4.3.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_WrapTight) is located in §A.4.4. *end note*]

20.4.2.20 wrapTopAndBottom (Top and Bottom Wrapping)

This element specifies that text shall wrap around the top and bottom of this object, but not its left or right edges.

[*Example*: Consider a DrawingML object using top and bottom wrapping and defined as follows:

```
<wp:anchor … >
  …
  <wp:wrapTopAndBottom />
</wp:anchor>
```

The wrapTopAndBottom element specifies that text must wrap around neither side of this object. *end example*]

Attributes	Description

distB (Distance From Text on Bottom Edge)	Specifies the minimum distance which shall be maintained between the bottom edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents.
	
	The distance shall be measured in EMUs (English Metric Units).
	
	[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its bottom edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor … >`
	`  …`
	`  <wp:wrapTopAndBottom distB="457200" … />`
	`</wp:anchor>`
	
	The distB attribute specifies that the padding distance must be `457200` EMUs or one-half of an inch. *end example*]
	
	The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

distT (Distance From Text on Top Edge)	Specifies the minimum distance which shall be maintained between the top edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents. The distance shall be measured in EMUs (English Metric Units). [*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its top edge and the nearest text. This setting would be specified as follows: `<wp:anchor … >`

Attributes	Description
	… <wp:wrapTopAndBottom distT="457200" … /> </wp:anchor> The distT attribute specifies that the padding distance must be 457200 EMUs or one-half of an inch. *end example*] The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

[*Note*: The W3C XML Schema definition of this element's content model (CT_WrapTopBottom) is located in §A.4.4. *end note*]

20.4.2.21 bg (Background Formatting)

This element defines formatting that can be applied to the background shape of the document. The background shape can hold formatting options just as a normal shape can hold within DrawingML.

20.4.2.22 bodyPr (Body Properties)

This element defines the body properties for the text body within a shape.

Attributes	Description
anchor (Anchor)	Specifies the anchoring position of the txBody within the shape. If this attribute is omitted, then a value of t, or top is implied. The possible values for this attribute are defined by the ST_TextAnchoringType simple type (§20.1.10.59).
anchorCtr (Anchor Center)	Specifies the centering of the text box. The way it works fundamentally is to determine the smallest possible "bounds box" for the text and then to center that "bounds box" accordingly. This is different than paragraph alignment, which aligns the text within the "bounds box" for the text. This flag is compatible with all of the different kinds of anchoring. If this attribute is omitted, then a value of 0 or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
bIns (Bottom Inset)	Specifies the bottom inset of the bounding rectangle. Insets are used just as internal margins for text boxes within shapes. If this attribute is omitted, a value of 45720 or 0.05 inches is implied. The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
compatLnSpc (Compatible Line	Specifies that the line spacing for this text body is decided in a simplistic manner using the font scene. If this attribute is omitted, a value of 0 or false is implied.

Attributes	Description
Spacing)	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
forceAA (Force Anti-Alias)	Forces the text to be rendered anti-aliased regardless of the font size. Certain fonts can appear grainy around their edges unless they are anti-aliased. Therefore this attribute allows for the specifying of which bodies of text should always be anti-aliased and which ones should not. If this attribute is omitted, then a value of `0` or `false` is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fromWordArt (From WordArt)	Specifies that text within this textbox is converted text from a WordArt object. This is more of a backwards compatibility attribute that is useful to the application from a tracking perspective. WordArt was the former way to apply text effects and therefore this attribute is useful in document conversion scenarios. If this attribute is omitted, then a value of `0` or `false` is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
horzOverflow (Text Horizontal Overflow)	Determines whether the text can flow out of the bounding box horizontally. This is used to determine what happens in the event that the text within a shape is too large for the bounding box it is contained within. If this attribute is omitted, then a value of overflow is implied. The possible values for this attribute are defined by the ST_TextHorzOverflowType simple type (§20.1.10.68).
lIns (Left Inset)	Specifies the left inset of the bounding rectangle. Insets are used just as internal margins for text boxes within shapes. If this attribute is omitted, then a value of 91440 or 0.1 inches is implied. The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
numCol (Number of Columns)	Specifies the number of columns of text in the bounding rectangle. When applied to a text run this property takes the width of the bounding box for the text and divides it by the number of columns specified. These columns are then treated as overflow containers in that when the previous column has been filled with text the next column acts as the repository for additional text. When all columns have been filled and text still remains then the overflow properties set for this text body are used and the text is reflowed to make room for additional text. If this attribute is omitted, then a value of 1 is implied. The possible values for this attribute are defined by the ST_TextColumnCount simple type (§20.1.10.64).
rIns (Right Inset)	Specifies the right inset of the bounding rectangle. Insets are used just as internal margins for text boxes within shapes. If this attribute is omitted, then a value of 91440 or 0.1 inches is implied.

Attributes	Description
	The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
rot (Rotation)	Specifies the rotation that is being applied to the text within the bounding box. If it not specified, the rotation of the accompanying shape is used. If it is specified, then this is applied independently from the shape. That is the shape can have a rotation applied in addition to the text itself having a rotation applied to it. If this attribute is omitted, then a value of 0, is implied. The possible values for this attribute are defined by the ST_Angle simple type (§20.1.10.3).
rtlCol (Columns Right-To-Left)	Specifies whether columns are used in a right-to-left or left-to-right order. The usage of this attribute only sets the column order that is used to determine which column overflow text should go to next. If this attribute is omitted, then a value of `0` or `false`is implied in which case text starts in the leftmost column and flow to the right. [*Note*: This attribute in no way determines the direction of text but merely the direction in which multiple columns are used. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
spcCol (Space Between Columns)	Specifies the space between text columns in the text area. This should only apply when there is more than 1 column present. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_PositiveCoordinate32 simple type (§20.1.10.42).
spcFirstLastPara (Paragraph Spacing)	Specifies whether the before and after paragraph spacing defined by the user is to be respected. While the spacing between paragraphs is helpful, it is additionally useful to be able to set a flag as to whether this spacing is to be followed at the edges of the text body, in other words the first and last paragraphs in the text body. More precisely since this is a text body level property it should only effect the before paragraph spacing of the first paragraph and the after paragraph spacing of the last paragraph for a given text body. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
tIns (Top Inset)	Specifies the top inset of the bounding rectangle. Insets are used just as internal margins for text boxes within shapes. If this attribute is omitted, then a value of 45720 or 0.05 inches is implied. The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
upright (Text	Specifies whether text should remain upright, regardless of the transform applied to it

Attributes	Description
Upright)	and the accompanying shape transform. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
vert (Vertical Text)	Determines if the text within the given text body should be displayed vertically. If this attribute is omitted, then a value of horz, or no vertical text is implied. The possible values for this attribute are defined by the ST_TextVerticalType simple type (§20.1.10.82).
vertOverflow (Text Vertical Overflow)	Determines whether the text can flow out of the bounding box vertically. This is used to determine what happens in the event that the text within a shape is too large for the bounding box it is contained within. If this attribute is omitted, then a value of overflow is implied. The possible values for this attribute are defined by the ST_TextVertOverflowType simple type (§20.1.10.83).
wrap (Text Wrapping Type)	Specifies the wrapping options to be used for this text body. If this attribute is omitted, then a value of square is implied which wraps the text using the bounding text box. The possible values for this attribute are defined by the ST_TextWrappingType simple type (§20.1.10.84).

20.4.2.23 cNvCnPr (Non-Visual Connector Shape Drawing Properties)

This element specifies the non-visual drawing properties specific to a connector shape. This includes information specifying the shapes to which the connector shape is connected.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualConnectorProperties) is located in §A.4.4. *end note*]

20.4.2.24 cNvContentPartPr (Non-Visual Content Part Drawing Properties)

This element specifies the non-visual drawing properties for a content part. This allows for additional information that does not affect the appearance of the content part to be stored.

Attributes	Description
isComment (Is a Comment) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies whether the content part is a comment or an annotation. If `true`, it is a comment; otherwise, it is a general annotation. The default value for this attribute is `true`. [*Example*: Consider a WordprocessingDrawingML object defined as follows: `<… isComment="false">`

Attributes	Description
	The content part shape does not represent a comment. *end example*] 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_NonVisualContentPartProperties) is located in §A.4.1. *end note*]

20.4.2.25 cNvFrPr (Non-Visual Graphic Frame Drawing Properties)

This element specifies the non-visual drawing properties for a graphic frame. These non-visual properties are properties that the generating application would utilize when rendering.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualGraphicFrameProperties) is located in §A.4.1. *end note*]

20.4.2.26 cNvGrpSpPr (Non-Visual Group Shape Drawing Properties)

This element specifies the non-visual drawing properties for a group shape. These non-visual properties are properties that the generating application would utilize when rendering.

20.4.2.27 cNvPr (Non-Visual Drawing Properties)

This element specifies non-visual canvas properties. This allows for additional information that does not affect the appearance of the picture to be stored.

[*Example*: Consider the following WordprocessingDrawingML:

```
<wsp>
  …
  <cNvPr id="4" name="Lilly_by_Lisher.jpg"/>
  …
</wsp>
```

end example]

Attributes	Description
descr (Alternative Text for Object) Namespace: http://purl.oclc.org/ooxml/drawingml/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:

Attributes	Description
	<… 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: http://purl.oclc.org/ooxml/drawingml/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: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.21).
name (Name) Namespace: http://purl.oclc.or	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:

Attributes	Description
g/ooxml/drawingml/main	`< … 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: http://purl.oclc.org/ooxml/drawingml/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.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualDrawingProps) is located in §A.4.1. *end note*]

20.4.2.28 cNvSpPr (Non-Visual Drawing Properties for a Shape)

This element specifies the non-visual drawing properties for a shape. These properties are to be used by the generating application to determine how the shape should be dealt with.

Attributes	Description
txBox (Text Box) Namespace: http://purl.oclc.org/ooxml/drawingml/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.

20.4.2.29 contentPart (Content Part)

This element specifies a reference to XML content in a format not defined by ECMA-376. [*Note*: This part allows the native use of other commonly used interchange formats, such as:

- MathML (http://www.w3.org/TR/MathML2/)
- SMIL (http://www.w3.org/TR/REC-smil/)
- SVG (http://www.w3.org/TR/SVG11/)

end note]

The relationship type of the explicit relationship specified by this element shall be http://purl.oclc.org/ooxml/officeDocument/relationships/customXml and have a TargetMode attribute value of `Internal`. If an application cannot process content of the content type specified by the targeted part, then it should continue to process the file. If possible, it should also provide some indication that unknown content was not imported.

Attributes	Description
bwMode (Black and White Mode)	Specifies how to interpret color information contained within a content part to achieve a color, black and white, or grayscale rendering of the content part. This attribute specifies only the rendering mode applied to the content part; it does not affect how the actual color information is persisted. The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (§20.1.10.10).
id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: • http://purl.oclc.org/ooxml/officeDocument/customXml for the contentPart element • http://purl.oclc.org/ooxml/officeDocument/relationships/footer for the footerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/header for the headerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements • http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings for the printerSettings element • http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: <… r:id="rId10" /> 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 (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_WordprocessingContentPart) is located in §A.4.4. *end note*]

20.4.2.30 extLst (Extension List)

This element specifies an extension list, within which all future extensions are defined within ext elements.

The extension list along with corresponding future extensions is used to extend the storage capabilities of the DrawingML framework. This allows for various new types of data to be stored natively within the existing diagram syntax.

20.4.2.31 graphicFrame (Graphical object container)

This element specifies a container for a graphical object in WordprocessingML.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GraphicFrame) is located in §A.4.5. *end note*]

20.4.2.32 grpSp (Group Shape)

This element specifies a group shape that represents many shapes grouped together. This shape is to be treated just as if it were a regular shape but instead of being described by a single geometry it is made up of all the shape geometries encompassed within it. Within a group shape each of the shapes that make up the group are specified just as they normally would. The idea behind grouping elements however is that a single transform can apply to many shapes at the same time.

20.4.2.33 grpSpPr (Group Shape Properties)

This element specifies the properties that are to be common across all of the shapes within the corresponding group. If there are any conflicting properties within the group shape properties and the individual shape properties then the individual shape properties should take precedence.

Attributes	Description
bwMode (Black and White Mode) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.10).

20.4.2.34 linkedTxbx (Textual contents of shape)

This element specifies the textual contents of a shape that is not the first in the series of shapes for the same text box story.

Attributes	Description
id (ID)	Specifies the identity of the text box story begun by a txbx element. This value shall be unique across a document for each txbx element. The possible values for this attribute are defined by the W3C XML Schema unsignedShort datatype.
seq (sequence index)	Specifies the position of the owning shape in the given text box story. The possible values for this attribute are defined by the W3C XML Schema unsignedShort datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ LinkedTextboxInformation) is located in §A.4.4. *end note*]

20.4.2.35 spPr (Shape Properties)

This element specifies the visual shape properties that can be applied to a shape. These properties include the shape fill, outline, geometry, effects, and 3D orientation.

Attributes	Description
bwMode (Black and White Mode) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.10).

20.4.2.36 style (Shape Style)

This element specifies the style information for a shape. This is used to define a shape's appearance in terms of the preset styles defined by the style matrix for the theme.

20.4.2.37 txbx (Textual contents of shape)

This element specifies the textual contents of a shape which is the first in the series of shapes for the same text box story. This element shall be present only in the CT_WordprocessingShape element that is the first in a series of CT_WordprocessingShape elements that refer to the same text box story.

Attributes	Description
id (ID)	Specifies the identity of the text box story begun by a txbx element. This value shall be unique across a document for each txbx element. The possible values for this attribute are defined by the W3C XML Schema unsignedShort datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextboxInfo) is located in §A.4.4. *end note*]

20.4.2.38 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 DrawingML syntax.

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)
- Vector Markup Language (VML)
- Additional txbxContent elements (as part of nested DrawingML objects)

20.4.2.39 wgp (WordprocessingML Shape Group)

This element specifies a shape group in WordprocessingML.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ WordprocessingGroup) is located in §A.4.5. *end note*]

20.4.2.40 whole (Whole E2O Formatting)

Formatting that applies to the entire diagram object, and not just the background, includes line and effect properties.

20.4.2.41 wpc (WordprocessingML Drawing Canvas)

This element specifies a drawing canvas in WordprocessingML. A drawing canvas is a logical grouping of shapes. [*Note*: A Drawing Canvas is typically used to allow grouping of shapes together for bulk operations. *end note*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ WordprocessingCanvas) is located in §A.4.5. *end note*]

20.4.2.42 wsp (WordprocessingML Shape)

This element specifies a shape in WordprocessingML.

Attributes	Description
normalEastAsianFlow (East Asian Flow)	Specifies that the text flow of the text contents of the shape shall ignore the text flow value specified by the vert attribute of the bodyPr element. If this attribute is set to TRUE then the text flows in the manner specified by the value "tbV" for ST_TextDirection (§17.18.93). 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_ WordprocessingShape) is located in §A.4.5. *end note*]

20.4.2.43 xfrm (2D Transform for Graphic Frames)

This element specifies a two dimensional transform for a Graphic Frame.

Attributes	Description
flipH (Horizontal Flip) Namespace: http://purl.oclc.org/ooxml/drawingml/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: http://purl.oclc.org/ooxml/drawingml/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*]

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rot (Rotation) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.3).

20.4.3 Simple Types

This is the complete list of simple types dedicated to DrawingML – WordprocessingML Drawing.

20.4.3.1 ST_AlignH (Relative Horizontal Alignment Positions)

This simple type contains the possible settings specifying how a DrawingML object can be horizontally aligned relative to the horizontal alignment base defined by the parent element.

[*Example*: Consider a picture in a WordprocessingML document which has been aligned relative to the edge of the page - the left of the page horizontally, and the top of the page vertically. This alignment would be specified as follows:

```
<wp:anchor … >
  <wp:positionH relativeFrom="page">
    <wp:align>left</wp:align>
  </wp:positionH>
  …
</wp:anchor>
```

The align element with a value of `left` specifies that for the horizontal positioning defined by the parent element (in this case, positioning relative to the page), the picture must be aligned to the left edge of the page. *end example*]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
center (Center Alignment)	Specifies that the object shall be centered with respect to the horizontal alignment base. [*Example*: Centered on the page. *end example*]
inside (Inside)	Specifies that the object shall be inside of the horizontal alignment base.

Enumeration Value	Description
	[*Example*: Inside the outside margin. *end example*]
left (Left Alignment)	Specifies that the object shall be left aligned to the horizontal alignment base. [*Example*: Left aligned relative to the margins. *end example*]
outside (Outside)	Specifies that the object shall be outside of the horizontal alignment base. [*Example*: Outside the left margin. *end example*]
right (Right Alignment)	Specifies that the object shall be right aligned to the horizontal alignment base. [*Example*: Right aligned relative to the margins. *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AlignH) is located in §A.4.4. *end note*]

20.4.3.2 ST_AlignV (Vertical Alignment Definition)

This simple type contains the possible settings specifying how a DrawingML object can be vertically aligned relative to the vertical alignment base defined by the parent element.

[*Example*: Consider a picture in a WordprocessingML document which has been aligned relative to the edge of the page - the left of the page horizontally, and the top of the page vertically. This alignment would be specified as follows:

```
<wp:anchor … >
  <wp:positionV relativeFrom="page">
    <wp:align>top</wp:align>
  </wp:positionH>
  …
</wp:anchor>
```

The align element with a value of top specifies that for the vertical positioning defined by the parent element (in this case, positioning relative to the page), the picture must be aligned to the top edge of the page. *end example*]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bottom (Bottom)	Specifies that the object shall be at the bottom of the vertical alignment base. [*Example*: Bottom of the page. *end example*]
center (Center Alignment)	Specifies that the object shall be centered with respect to the vertical alignment base. [*Example*: Centered on the page. *end example*]
inside (Inside)	Specifies that the object shall be inside of the horizontal alignment base. [*Example*: Inside the top margin. *end example*]
outside (Outside)	Specifies that the object shall be outside of the vertical alignment base. [*Example*: Outside the top margin. *end example*]
top (Top)	Specifies that the object shall be at the top of the vertical alignment base. [*Example*: Top of the page. *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AlignV) is located in §A.4.4. *end note*]

20.4.3.3 ST_PositionOffset (Absolute Position Offset Value)

This simple type represents a one dimensional distance which shall be used to offset an objet from its base positioning location stored in EMUs.

[*Example*: Consider a DrawingML picture which must be displayed one inch from the top of the page, and one-half of an inch from the left edge of the page. This object would be specified as follows:

```
<wp:anchor … >
  <wp:positionH relativeFrom="page">
    <wp:posOffset>914400</wp:posOffset>
  </wp:positionH>
  <wp:positionV relativeFrom="page">
    <wp:posOffset>457200</wp:posOffset>
  </wp:positionV>
</wp:anchor>
```

The posOffset element specifies the absolute positioning of the object relative to the top-left edge of the page in EMUs. *end example*]

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_PositionOffset) is located in §A.4.4. *end note*]

20.4.3.4 ST_RelFromH (Horizontal Relative Positioning)

This simple type specifies the possible values for the base from which the relative horizontal positioning of an object shall be calculated.

[*Example*: Consider a DrawingML picture which must be displayed at the bottom center of the page. This object would be specified as follows:

```
<wp:anchor … >
  <wp:positionH relativeFrom="page">
    <wp:align>center</wp:align>
  </wp:positionH>
  …
</wp:anchor>
```

The relativeFrom attribute specifies that the object is horizontally positioned relative to the page. *end example*]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
character (Character)	Specifies that the horizontal positioning shall be relative to the position of the anchor within its run content.
column (Column)	Specifies that the horizontal positioning shall be relative to the extents of the column which contains its anchor.
insideMargin (Inside Margin)	Specifies that the horizontal positioning shall be relative to the inside margin of the current page (the left margin on odd pages, right on even pages).
leftMargin (Left Margin)	Specifies that the horizontal positioning shall be relative to the left margin of the page.
margin (Page Margin)	Specifies that the horizontal positioning shall be relative to the page margins.
outsideMargin (Outside Margin)	Specifies that the horizontal positioning shall be relative to the outside margin of the current page (the

Enumeration Value	Description
	right margin on odd pages, left on even pages).
page (Page Edge)	Specifies that the horizontal positioning shall be relative to the edge of the page.
rightMargin (Right Margin)	Specifies that the horizontal positioning shall be relative to the right margin of the page.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RelFromH) is located in §A.4.4. *end note*]

20.4.3.5 ST_RelFromV (Vertical Relative Positioning)

This simple type specifies the possible values for the base from which the relative vertical positioning of an object shall be calculated.

[*Example*: Consider a DrawingML picture which must be displayed at the bottom center of the page. This object would be specified as follows:

```
<wp:anchor … >
  <wp:positionV relativeFrom="page">
    <wp:align>bottom</wp:align>
  </wp:positionV>
  …
</wp:anchor>
```

The relativeFrom attribute specifies that the object is horizontally positioned relative to the page. *end example*]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bottomMargin (Bottom Margin)	Specifies that the vertical positioning shall be relative to the bottom margin of the current page.
insideMargin (Inside Margin)	Specifies that the vertical positioning shall be relative to the inside margin of the current page.
line (Line)	Specifies that the vertical positioning shall be relative to the line containing the anchor character.
margin (Page Margin)	Specifies that the vertical positioning shall be relative to the page margins.
outsideMargin (Outside Margin)	Specifies that the vertical positioning shall be relative to the outside margin of the current page.
page (Page Edge)	Specifies that the vertical positioning shall be relative

Enumeration Value	Description
	to the edge of the page.
paragraph (Paragraph)	Specifies that the vertical positioning shall be relative to the paragraph which contains the drawing anchor.
topMargin (Top Margin)	Specifies that the vertical positioning shall be relative to the top margin of the current page.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RelFromV) is located in §A.4.4. *end note*]

20.4.3.6 ST_WrapDistance (Distance from Text)

This simple type represents a one dimensional distance which shall be used to offset an object from text, stored in EMUs.

[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its left edge and the nearest text. This setting would be specified as follows:

```
<wp:anchor … >
  …
  <wp:wrapThrough distL="457200" … />
</wp:anchor>
```

The distL attribute specifies that the padding distance must be `457200` EMUs or one-half of an inch. *end example*]

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_WrapDistance) is located in §A.4.4. *end note*]

20.4.3.7 ST_WrapText (Text Wrapping Location)

This simple type specifies the possible settings for how text can 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 would be specified as follows:

```
<wp:anchor … >
  …
  <wp:wrapTight wrapText="left" … />
</wp:anchor>
```

The wrapText attribute value of `left` specifies that text must only wrap around the let side of the object. *end example*]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bothSides (Both Sides)	Specifies that text shall wrap around both sides of the object.
largest (Largest Side Only)	Specifies that text shall only wrap around the largest side of the object. If the object is positioned in the exact center of the page, the text shall wrap around the side on which text is first encountered: • If the first line of text intersecting the object is using left-to-right reading order, the text shall wrap to the object's left. • If the first line of text intersecting the object is using right-to-left reading order, the text shall wrap to the object's right.
left (Left Side Only)	Specifies that text shall only wrap around the left side of the object.
right (Right Side Only)	Specifies that text shall only wrap around the right side of the object.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_WrapText) is located in §A.4.4. *end note*]

20.5 DrawingML - SpreadsheetML Drawing

Within a SpreadsheetML document, it is possible to include graphical DrawingML objects:

- Pictures (§20.2)
- Locked Canvases (§20.3)
- Diagrams (§21.4)
- Charts (§21.2)

When these objects are present in a spreadsheet document, it is necessary to include information which specifies how the objects shall be positioned relative to the parent worksheet. [*Example*: Whether the object is anchored to a specific row, whether it resizes with cells, and so on. *end example*]

The SpreadsheetML Drawing namespace acts in this capacity, specifying all information necessary to anchor and display DrawingML objects within a spreadsheet document.

[*Example*: Consider a DrawingML picture which must be anchored to a specific cell for its top left and bottom right corners, resizing as those cells are relocated. This object would be specified as follows:

```
<xdr:twoCellAnchor>
  <xdr:from>
    …
  </xdr:from>
  <xdr:to>
    …
  </xdr:to>
  <xdr:graphicFrame>
    …
    <a:graphic>
      <a:graphicData
         uri="http://purl.oclc.org/ooxml/drawingml/diagram">
        <dgm:relIds xmlns:dgm="…" xmlns:r="…" r:dm="rId1" r:lo="rId2"
           r:qs="rId3" r:cs="rId4" />
      </a:graphicData>
    </a:graphic>
  </xdr:graphicFrame>
</xdr:twoCellAnchor>
```

The twoCellAnchor element (§20.5.2.33) specifies that this object anchored to the cells specified by the to (§20.5.2.32) and from (§20.5.2.15) elements. *end example*]

20.5.1 Table of Contents

This subclause is informative.

End of informative text.

20.5.2 Elements

The following elements define the contents of the Spreadsheet Drawing namespace:

20.5.2.1 absoluteAnchor (Absolute Anchor Shape Size)

This element is used as an anchor placeholder for a shape or group of shapes. It anchors the object in the same position relative to sheet position and its extents are in EMU units.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AbsoluteAnchor) is located in §A.4.5. *end note*]

20.5.2.2 blipFill (Picture Fill)

This element specifies the type of picture fill that the picture object has. Because a picture has a picture fill already by default, it is possible to have two fills specified for a picture object. An example of this is shown below.

[*Example*: Consider the picture below that has a blip fill applied to it. The image used to fill this picture object has transparent pixels instead of white pixels.

```
<xdr:pic>
  ..
```

```
    <xdr:blipFill>
      <a:blip r:embed="rId2"/>
      <a:stretch>
        <a:fillRect/>
      </a:stretch>
    </xdr:blipFill>
    ..
  </xdr:pic>
```

The above picture object is shown as an example of this fill type. *end example*]

[*Example*: Consider now the same picture object but with an additional gradient fill applied within the shape properties portion of the picture.

```
  <xdr:pic>
    ..
    <xdr:blipFill>
      <a:blip r:embed="rId2"/>
      <a:stretch>
        <a:fillRect/>
      </a:stretch>
    </xdr:blipFill>
    <xdr:spPr>
      <a:gradFill>
        <a:gsLst>
          <a:gs pos="0">
            <a:schemeClr val="tx2">
              <a:shade val="50000"/>
            </a:schemeClr>
          </a:gs>
          <a:gs pos="39999">
            <a:schemeClr val="tx2">
```



```
            <a:tint val="20000"/>
          </a:schemeClr>
        </a:gs>
        <a:gs pos="70000">
          <a:srgbClr val="C4D6EB"/>
        </a:gs>
        <a:gs pos="100000">
          <a:schemeClr val="bg1"/>
        </a:gs>
      </a:gsLst>
    </a:gradFill>
  </xdr:spPr>
  ..
</xdr:pic>
```

The above picture object is shown as an example of this double fill type. *end example*]

Attributes	Description
dpi (DPI Setting) Namespace: http://purl.oclc.org/ooxml/drawingml/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: http://purl.oclc.or	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

Attributes	Description
g/ooxml/drawingml/main	datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BlipFillProperties) is located in §A.4.1. *end note*]

20.5.2.3 clientData (Client Data)

This element is used to set certain properties related to a drawing element on the client spreadsheet application.

Attributes	Description
fLocksWithSheet (Locks With Sheet Flag)	This attribute indicates whether to disable selection on drawing elements when the sheet is protected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fPrintsWithSheet (Prints With Sheet Flag)	This attribute indicates whether to print drawing elements when printing the sheet. 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_AnchorClientData) is located in §A.4.5. *end note*]

20.5.2.4 cNvCxnSpPr (Non-Visual Connector Shape Drawing Properties)

This element specifies the non-visual properties for a connector shape. These are the set of properties on a shape which do not affect its display within a spreadsheet.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualConnectorProperties) is located in §A.4.1. *end note*]

20.5.2.5 cNvGraphicFramePr (Non-Visual Graphic Frame Drawing Properties)

This element specifies the non-visual properties for a single graphical object frame within a spreadsheet. These are the set of properties of a frame which do not affect its display within a spreadsheet.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualGraphicFrameProperties) is located in §A.4.1. *end note*]

20.5.2.6 cNvGrpSpPr (Non-Visual Group Shape Drawing Properties)

This element specifies the non-visual properties of a hierarchical grouping of shapes, graphical object frames, and child groups. These are the set of properties of a group which do not affect its display within a spreadsheet.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualGroupDrawingShapeProps) is located in §A.4.1. *end note*]

20.5.2.7 cNvPicPr (Non-Visual Picture Drawing Properties)

This element describes the non-visual properties of a picture within a spreadsheet. These are the set of properties of a picture which do not affect its display within a spreadsheet.

[*Example*: Consider the following SpreadsheetDrawingML.

```
<xdr:pic>
  ..
  <xdr:nvPicPr>
    <xdr:cNvPr id="4" name="Lilly_by_Lisher.jpg"/>
    <xdr:cNvPicPr>
      <a:picLocks noChangeAspect="1"/>
    </xdr:cNvPicPr>
    <xdr:nvPr/>
  </xdr:nvPicPr>
  ..
</xdr:pic>
```

The above example defines some non-visual picture drawing properties for the inserted picture. *end example*]

Attributes	Description
preferRelativeResize (Relative Resize Preferred) Namespace: http://purl.oclc.org/ooxml/drawingml/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.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualPictureProperties) is located in §A.4.1. *end note*]

20.5.2.8 cNvPr (Non-Visual Drawing Properties)

This element specifies the set of non-visual properties for the parent element. These properties specify all the data about the parent which does not affect its display within the spreadsheet.

[*Example*: Consider the following SpreadSheetDrawingML.

```
<xdr:pic>
  ..
  <xdr:nvPicPr>
    <xdr:cNvPr id="4" name="Lilly_by_Lisher.jpg"/>
  </xdr:nvPicPr>
  ..
</xdr:pic>
```

The above example defines some non-visual drawing properties for the inserted picture. *end example*]

Attributes	Description
descr (Alternative Text for Object) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies alternative text for the current DrawingML object, for use by assistive technologies or applications which 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 which 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: http://purl.oclc.org/ooxml/drawingml/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 which 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*]

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Unique Identifier) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.21).
name (Name) Namespace: http://purl.oclc.org/ooxml/drawingml/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: http://purl.oclc.org/ooxml/drawingml/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.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualDrawingProps) is located in §A.4.1. *end note*]

20.5.2.9 cNvSpPr (Connection Non-Visual Shape Properties)

This element specifies the set of non-visual properties for a connection shape. These properties specify all data about the connection shape which do not affect its display within a spreadsheet.

[*Example*: Consider the shape that has a shape lock applied to it.

```
<xdr:sp>
  <xdr:nvSpPr>
    <xdr:cNvPr id="2" name="Rectangle 1"/>
    <xdr:cNvSpPr>
      <a:spLocks noGrp="1"/>
    </xdr:cNvSpPr>
   </xdr:nvSpPr>
   ..
</xdr:sp>
```

This shape lock is stored within the non-visual drawing properties for this shape. *end example*]

Attributes	Description
txBox (Text Box) Namespace: http://purl.oclc.org/ooxml/drawingml/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.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualDrawingShapeProps) is located in §A.4.1. *end note*]

20.5.2.10 col (Column))

This element specifies the column that is used within the from and to elements to specify anchoring information for a shape within a spreadsheet

The possible values for this element are defined by the ST_ColID simple type (§20.5.3.1).

[*Note*: The W3C XML Schema definition of this element's content model (ST_ColID) is located in §A.4.5. *end note*]

20.5.2.11 colOff (Column Offset)

This element is used to specify the column offset within a cell. The units for which this attribute is specified in reside within the simple type definition referenced below.

The possible values for this element are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Coordinate) is located in §A.4.1. *end note*]

20.5.2.12 contentPart (Content Part)

This element specifies a reference to XML content in a format not defined by ISO/IEC 29500. [*Note*: This part allows the native use of other commonly used interchange formats, such as:

- MathML (http://www.w3.org/TR/MathML2/)
- SMIL (http://www.w3.org/TR/REC-smil/)
- SVG (http://www.w3.org/TR/SVG11/)

end note]

The relationship type of the explicit relationship specified by this element shall be http://purl.oclc.org/ooxml/officeDocument/relationships/customXml and have a TargetMode attribute value of `Internal`. If an application cannot process content of the content type specified by the targeted part, then it should continue to process the file. If possible, it should also provide some indication that unknown content was not imported.

[*Example*: Consider a SpreadsheetML document which includes the following SVG markup in a part named svg1.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<!--======================================================================-->
<!--=  Copyright 2000 World Wide Web Consortium, (Massachusetts          =-->
<!--=  Institute of Technology, Institut National de Recherche en        =-->
<!--=  Informatique et en Automatique, Keio University). All Rights      =-->
<!--=  Reserved. See http://www.w3.org/Consortium/Legal/.                =-->
<!--======================================================================-->
<!-- ====================================================================-->
<!--                                                                     -->
<!-- color-datatypes-BE-01.svg                                           -->
<!-- renamed for 1.1 suite to color-prop-02-f.svg                        -->
<!--                                                                     -->
<!-- Author : Chris Lilley,  12-Aug-2000                                 -->
<!--          1.1 revision by Rick Graham                                -->
```

```
<!--          Revised for SVGT/B: Benoit Bezaire Jul/02/2002             -->
<!--          More revision  CL  -->
<!--=======================================================================-->
<svg version="1.1" baseProfile="full" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" id="svg-root" width="100%"
height="100%" viewBox="0 0 480 360">
  <title id="test-title">color-prop-02-f.svg</title>
  <desc id="test-desc">Test that viewer has the basic capability to render
X11colors, using any of the equivalent forms.</desc>
  <!--=================================================================-->
  <!--Content of Test Case follows…                ===============-->
  <!--=================================================================-->
  <g id="test-body-content">
    <!-- groups of five colors  -->
    <g>
      <circle cx="75" cy="40" r="20" fill="crimson"/>
      <circle cx="115" cy="40" r="20" fill="#DC143C"/>
      <circle cx="75" cy="80" r="20" fill="rgb(220,20,60)"/>
      <circle cx="115" cy="80" r="20"
fill="rgb(86.27450980392156862745098039215%,7.84313725490196078431372549019%,
23.52941176470588235294117647058%)"/>
    </g>
    <g>
      <circle cx="200" cy="40" r="20" fill="palegreen"/>
      <circle cx="240" cy="40" r="20" fill="#98FB98"/>
      <circle cx="200" cy="80" r="20" fill="rgb(152, 251, 152)"/>
      <circle cx="240" cy="80" r="20"
fill="rgb(59.607843137254901960784313725490%,98.431372549019607843137254901961%,5
9.607843137254901960784313725490%)"/>
    </g>
    <g>
      <circle cx="325" cy="40" r="20" fill="royalblue"/>
      <circle cx="365" cy="40" r="20" fill="#4169E1"/>
      <circle cx="325" cy="80" r="20" fill="rgb(65, 105, 225)"/>
      <circle cx="365" cy="80" r="20"
fill="rgb(25.490196078431372549019607843137%,41.176470588235294117647058823529%,
88.235294117647058823529411764706%)"/>
    </g>
    <g>
      <circle cx="75" cy="135" r="20" fill="firebrick"/>
      <circle cx="115" cy="135" r="20" fill="#B22222"/>
      <circle cx="75" cy="175" r="20" fill="rgb(178,34,34)"/>
```



```
      <circle cx="115" cy="175" r="20"
fill="rgb(69.80392156862745098039215686274509%,13.333333333333333333333333333333%,
13.333333333333333333333333333333%)"/>
    </g>
    <g>
      <circle cx="200" cy="135" r="20" fill="seagreen"/>
      <circle cx="240" cy="135" r="20" fill="#2E8B57"/>
      <circle cx="200" cy="175" r="20" fill="rgb(46, 139, 87)"/>
      <circle cx="240" cy="175" r="20"
fill="rgb(18.039215686274509803921568627451%,54.509803921568627450980392156863%,
34.117647058823529411764705882353%)"/>
    </g>
    <g>
      <circle cx="325" cy="135" r="20" fill="mediumblue"/>
      <circle cx="365" cy="135" r="20" fill="#0000CD"/>
      <circle cx="325" cy="175" r="20" fill="rgb(0, 0, 205)"/>
      <circle cx="365" cy="175" r="20"
fill="rgb(0%,0%,80.392156862745098039215686274509%)"/>
    </g>
    <g>
      <circle cx="75" cy="230" r="20" fill="indianred"/>
      <circle cx="115" cy="230" r="20" fill="#CD5C5C"/>
      <circle cx="75" cy="270" r="20" fill="rgb(205, 92, 92)"/>
      <circle cx="115" cy="270" r="20"
fill="rgb(80.392156862745098039215686274509%,36.078431372549019607843137254902%,3
6.078431372549019607843137254902%)"/>
    </g>
    <g>
      <circle cx="200" cy="230" r="20" fill="lawngreen"/>
      <circle cx="240" cy="230" r="20" fill="#7CFC00"/>
      <circle cx="200" cy="270" r="20" fill="rgb(124, 252, 0)"/>
      <circle cx="240" cy="270" r="20"
fill="rgb(48.627450980392156862745098039216%,98.823529411764705882352941176471%,
0%)"/>
    </g>
    <g>
      <circle cx="325" cy="230" r="20" fill="mediumturquoise"/>
      <circle cx="365" cy="230" r="20" fill="#48D1CC"/>
      <circle cx="325" cy="270" r="20" fill="rgb(72, 209, 204)"/>
      <circle cx="365" cy="270" r="20"
fill="rgb(28.235294117647058823529411764706%,81.960784313725490196078431372549%,
80%)"/>
    </g>
```

```
  </g>
  <text id="revision" x="10" y="340" font-size="40" stroke="none"
fill="black">$Revision: 1.6 $</text>
  <rect id="test-frame" x="1" y="1" width="478" height="358" fill="none"
stroke="#000000"/>
</svg>
```

The SpreadsheetML Drawing part would reference this content as follows:

```
<wsDr>
  <twoCellAnchor>
    <from>
      <col>3</col>
      <colOff>152400</colOff>
      <row>5</row>
      <rowOff>123825</rowOff>
    </from>
    <to>
      <col>8</col>
      <colOff>266700</colOff>
      <row>22</row>
      <rowOff>38100</rowOff>
    </to>
  </twoCellAnchor>
  <contentPart r:id="svg1"/>
</wsDr>
```

The contentPart element specifies that the SVG markup targeted by the relationship with an ID of `svg1` is part of the SpreadsheetML document. Examining the contents of the corresponding relationship part item, we can see the targets for that relationship:

```
<Relationships … >
  …
  <Relationship Id="svg1" TargetMode="Internal"
Type="http://purl.oclc.org/ooxml/officeDocument/relationships/customXml"
Target="svg1.xml" />
  …
</Relationships>
```

The corresponding relationship part item shows that the file to be imported is named `svg1.xml`. *end example*]

Attributes	Description
id (Relationship to Part)	Specifies the relationship ID to a content part. [*Example*: Consider an XML element which has the following id attribute:

Attributes	Description
Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	<… 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 (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rel) is located in §A.4.5. *end note*]

20.5.2.13 cxnSp (Connection Shape)

This element specifies the properties for a connection shape drawing element. A connection shape is a line, etc. that connects two other shapes in this drawing.

Attributes	Description
fPublished (Publish to Server Flag)	This attribute indicates whether the shape shall be published with the worksheet when sent to the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
macro (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*] The format of this string shall be application-defined, and should be ignored if not understood. [*Example*: < … macro="DoWork()" /> *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_Connector) is located in §A.4.5. *end note*]

20.5.2.14 ext (Shape Extent)

This element describes the length and width properties for how far a drawing element should extend for.

Attributes	Description
cx (Extent Length) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.41).
cy (Extent Width) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositiveSize2D) is located in §A.4.1. *end note*]

20.5.2.15 from (Starting Anchor Point)

This element specifies the first anchor point for the drawing element. This is used to anchor the top and left sides of the shape within the spreadsheet. That is when the cell that is specified in the from element is adjusted, the shape is also adjusted.

[*Example*: Consider the following SpreadsheetDrawingML

```
<xdr:twoCellAnchor>
  <xdr:from>
    <xdr:col>3</xdr:col>
    <xdr:colOff>447675</xdr:colOff>
    <xdr:row>8</xdr:row>
    <xdr:rowOff>28575</xdr:rowOff>
  </xdr:from>
  <xdr:to>
```



```
      <xdr:col>5</xdr:col>
      <xdr:colOff>466725</xdr:colOff>
      <xdr:row>14</xdr:row>
      <xdr:rowOff>9525</xdr:rowOff>
    </xdr:to>
    <xdr:sp macro="" textlink="">
    …
    </xdr:sp>
    <xdr:clientData/>
  </xdr:twoCellAnchor>
```

The above example shows the first anchor point being specified via the from element. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Marker) is located in §A.4.5. *end note*]

20.5.2.16 graphicFrame (Graphic Frame)

This element describes a single graphical object frame for a spreadsheet which contains a graphical object.

Attributes	Description
fPublished (Publish to Server Flag)	This attribute indicates whether the shape shall be published with the worksheet when sent to the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
macro (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*] The format of this string shall be application-defined, and should be ignored if not understood. [*Example*: `< … macro="DoWork()" />` *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_GraphicalObjectFrame) is located in §A.4.5. *end note*]

20.5.2.17 grpSp (Group Shape)

This element specifies a group shape that represents many shapes grouped together. This shape is to be treated just as if it were a regular shape but instead of being described by a single geometry it is made up of all the shape geometries encompassed within it. Within a group shape each of the shapes that make up the group are specified just as they normally would. The idea behind grouping elements however is that a single transform can apply to many shapes at the same time.

[*Example*: Consider the following group shape.

```
<xdr:grpSp>
  <xdr:nvGrpSpPr>
    <xdr:cNvPr id="10" name="Group 9"/>
    <xdr:cNvGrpSpPr/>
    <xdr:nvPr/>
  </xdr:nvGrpSpPr>
  <xdr:grpSpPr>
    <a:xfrm>
      <a:off x="838200" y="990600"/>
      <a:ext cx="2426208" cy="978408"/>
      <a:chOff x="838200" y="990600"/>
      <a:chExt cx="2426208" cy="978408"/>
   </a:xfrm>
  </xdr:grpSpPr>
  <xdr:sp>
  ..
  </xdr:sp>
  <xdr:sp>
  ..
  </xdr:sp>
  <xdr:sp>
  ..
  </xdr:sp>
</xdr:grpSp>
```

In the above example we see three shapes specified within a single group. These three shapes have their position and sizes specified just as they normally would within the shape tree. The generating application should apply the transformation after the bounding box for the group shape has been calculated. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupShape) is located in §A.4.5. *end note*]

20.5.2.18 grpSpPr (Group Shape Properties)

This element specifies the properties that are to be common across all of the shapes within the corresponding group. If there are any conflicting properties within the group shape properties and the individual shape properties then the individual shape properties should take precedence.

Attributes	Description
bwMode (Black and White Mode) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupShapeProperties) is located in §A.4.1. *end note*]

20.5.2.19 nvCxnSpPr (Non-Visual Properties for a Connection Shape)

This element specifies all non-visual properties for a connection shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a connection shape. This allows for additional information that does not affect the appearance of the connection shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ConnectorNonVisual) is located in §A.4.5. *end note*]

20.5.2.20 nvGraphicFramePr (Non-Visual Properties for a Graphic Frame)

This element specifies all non-visual properties for a graphic frame. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a graphic frame. This allows for additional information that does not affect the appearance of the graphic frame to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GraphicalObjectFrameNonVisual) is located in §A.4.5. *end note*]

20.5.2.21 nvGrpSpPr (Non-Visual Properties for a Group Shape)

This element specifies all non-visual properties for a group shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a group

shape. This allows for additional information that does not affect the appearance of the group shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupShapeNonVisual) is located in §A.4.5. *end note*]

20.5.2.22 nvPicPr (Non-Visual Properties for a Picture)

This element specifies all non-visual properties for a picture. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a picture. This allows for additional information that does not affect the appearance of the picture to be stored.

[*Example*: Consider the following SpreadsheetDrawingML.

```
<xdr:pic>
  ..
  <xdr:nvPicPr>
    ..
  </xdr:nvPicPr>
  ..
</xdr:pic>
```

The above example shows the defining of non-visual picture properties. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_PictureNonVisual) is located in §A.4.5. *end note*]

20.5.2.23 nvSpPr (Non-Visual Properties for a Shape)

This element specifies all non-visual properties for a shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a shape. This allows for additional information that does not affect the appearance of the shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeNonVisual) is located in §A.4.5. *end note*]

20.5.2.24 oneCellAnchor (One Cell Anchor Shape Size)

This element specifies a one cell anchor placeholder for a group, a shape, or a drawing element. It moves with the cell and its extents is in EMU units.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OneCellAnchor) is located in §A.4.5. *end note*]

20.5.2.25 pic (Picture)

This element specifies the existence of a picture object within the spreadsheet.

[*Example*: Consider the following SpreadsheetDrawingML that specifies the existence of a picture within a document. This picture can have non-visual properties, a picture fill as well as shape properties attached to it.

```
<xdr:pic>
  <xdr:nvPicPr>
    <xdr:cNvPr id="4" name="lake.JPG" descr="Picture of a Lake" />
    <xdr:cNvPicPr>
      <a:picLocks noChangeAspect="1"/>
    </xdr:cNvPicPr>
    <xdr:nvPr/>
  </xdr:nvPicPr>
  <xdr:blipFill>
  …
  </xdr:blipFill>
  <xdr:spPr>
  …
  </xdr:spPr>
</xdr:pic>
```

end example]

Attributes	Description
fPublished (Publish to Server Flag)	This attribute indicates whether the shape shall be published with the worksheet when sent to the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
macro (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*] The format of this string shall be application-defined, and should be ignored if not understood. [*Example*: `< … macro="DoWork()" />` *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_Picture) is located in §A.4.5. *end note*]

20.5.2.26 pos (Position)

This element describes the position of a drawing element within a spreadsheet.

Attributes	Description
x (X-Axis Coordinate) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.16).
y (Y-Axis Coordinate) Namespace: http://purl.oclc.org/ooxml/drawingml/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 y attribute defines a y-coordinate of `100`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Point2D) is located in §A.4.1. *end note*]

20.5.2.27 row (Row)

This element specifies the row that is used within the from and to elements to specify anchoring information for a shape within a spreadsheet.

The possible values for this element are defined by the ST_RowID simple type (§20.5.3.3).

[*Note*: The W3C XML Schema definition of this element's content model (ST_RowID) is located in §A.4.5. *end note*]

20.5.2.28 rowOff (Row Offset)

This element is used to specify the row offset within a cell. The units for which this attribute is specified in reside within the simple type definition referenced below.

The possible values for this element are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Coordinate) is located in §A.4.1. *end note*]

20.5.2.29 sp (Shape)

This element specifies the existence of a single shape. A shape can either be a preset or a custom geometry, defined using the SpreadsheetDrawingML framework. In addition to a geometry each shape can have both visual and non-visual properties attached. Text and corresponding styling information can also be attached to a shape. This shape is specified along with all other shapes within either the shape tree or group shape elements.

Attributes	Description
fLocksText (Lock Text Flag)	This attribute indicates whether to allow text editing within this drawing object when the parent worksheet is protected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fPublished (Publish to Server Flag)	This attribute indicates whether the shape shall be published with the worksheet when sent to the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
macro (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*] The format of this string shall be application-defined, and should be ignored if not understood. [*Example*: `< … macro="DoWork()" />` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
textlink (Text Link)	This attribute specifies a formula linking to spreadsheet cell data. 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.4.5. *end note*]

20.5.2.30 spPr (Shape Properties)

This element specifies the visual shape properties that can be applied to a special shape such as a connector shape or picture. These are the same properties that are allowed to describe the visual properties of a shape but are used here to describe additional object-specific properties within a document. This allows for these shapes to have both the properties of a shape as well as specific properties that are unique to only them.

Attributes	Description
bwMode (Black and White Mode) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeProperties) is located in §A.4.1. *end note*]

20.5.2.31 style (Shape Style)

The element specifies the style that is applied to a shape and the corresponding references for each of the style components such as lines and fills.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeStyle) is located in §A.4.1. *end note*]

20.5.2.32 to (Ending Anchor Point)

This element specifies the second anchor point for the drawing element. This is used to anchor the bottom and right sides of the shape within the spreadsheet. That is when the cell that is specified in the to element is adjusted, the shape is also adjusted.

[*Example*: Consider the following SpreadsheetDrawingML

```
<xdr:twoCellAnchor>
  <xdr:from>
    <xdr:col>3</xdr:col>
    <xdr:colOff>447675</xdr:colOff>
    <xdr:row>8</xdr:row>
    <xdr:rowOff>28575</xdr:rowOff>
```



```
    </xdr:from>
    <xdr:to>
      <xdr:col>5</xdr:col>
      <xdr:colOff>466725</xdr:colOff>
      <xdr:row>14</xdr:row>
      <xdr:rowOff>9525</xdr:rowOff>
    </xdr:to>
    <xdr:sp macro="" textlink="">
    …
    </xdr:sp>
    <xdr:clientData/>
  </xdr:twoCellAnchor>
```

The above example shows the second anchor point being specified via the to element. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Marker) is located in §A.4.5. *end note*]

20.5.2.33 twoCellAnchor (Two Cell Anchor Shape Size)

This element specifies a two cell anchor placeholder for a group, a shape, or a drawing element. It moves with cells and its extents are in EMU units.

Attributes	Description

editAs (Positioning and Resizing Behaviors)	Specifies how the DrawingML contents shall be moved and/or resized when the rows and columns between its start and ending anchor (the from and to child elements) are resized, or have additional rows/columns inserted within them, or additional row/columns are added before them. The behaviors are discussed in the simple type referenced below.
	
	If this attribute is omitted, then its default value shall be assumed to be `twoCell`.
	
	[*Example*: Consider a drawing defined as follows:
	
	`<ws:twoCellAnchor editAs="absolute">`
	`…`
	`</ws:twoCellAnchor>`
	
	The editAs attribute has a value of `absolute`, which specifies that the sizing of this object must not change, instead the anchor locations should be moved as needed to maintain the same size and position. *end example*]
	
	The possible values for this attribute are defined by the ST_EditAs simple type (§20.5.3.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwoCellAnchor) is located in §A.4.5. *end note*]

20.5.2.34 txBody (Shape Text Body)

This element specifies the existence of text to be contained within the corresponding shape. All visible text and visible text related properties are contained within this element. There can be multiple paragraphs and within paragraphs multiple runs of text.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBody) is located in §A.4.1. *end note*]

20.5.2.35 wsDr (Worksheet Drawing)

This element specifies all drawing objects within the worksheet. It acts much like the spTree element within the DrawingML framework. Allowing for the specification of all shapes for a given part of a document, in this case a single Worksheet.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Drawing) is located in §A.4.5. *end note*]

20.5.2.36 xfrm (2D Transform for Graphic Frames)

This element specifies a two dimensional transform for a Graphic Frame.

Attributes	Description
flipH (Horizontal Flip) Namespace: http://purl.oclc.org/ooxml/drawingml/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: http://purl.oclc.org/ooxml/drawingml/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.

Attributes	Description
	 *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rot (Rotation) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Transform2D) is located in §A.4.1. *end note*]

20.5.3 Simple Types

This is the complete list of simple types dedicated to DrawingML – SpreadsheetML Drawing.

20.5.3.1 ST_ColID (Column ID)

This simple type specifies a column identification. The numerical value used for the column id should be non-negative and never exceed the number of total columns within the spreadsheet document.

This simple type's contents are a restriction of the W3C XML Schema int datatype.

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_ColID) is located in §A.4.5. *end note*]

20.5.3.2 ST_EditAs (Resizing Behaviors)

This simple type specifies all possible settings for how DrawingML contents shall be resized when the rows and columns between its start and ending anchor (the from and to child elements) are resized, or have additional rows/columns inserted within them.

[*Example*: Consider a drawing defined as follows:

```
<ws:twoCellAnchor editAs="absolute">
  …
</ws:twoCellAnchor>
```

The editAs attribute has a value of absolute, which specifies that the sizing of this object must not change, instead the anchor locations should be moved as needed to maintain the same size. *end example*]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
absolute (Do Not Move or Resize With Underlying Rows/Columns)	Specifies that the current start and end positions shall be maintained with respect to the distances from the absolute start point of the worksheet. If additional rows/columns are added before the drawing, the drawing shall move its anchors as needed to maintain this same absolute position.
oneCell (Move With Cells but Do Not Resize)	Specifies that the current drawing shall move with its row and column (i.e. the object is anchored to the actual from row and column), but that the size shall remain absolute. If additional rows/columns are added between the from and to locations of the drawing, the drawing shall move its to anchors as needed to maintain this same absolute size.
twoCell (Move and Resize With Anchor Cells)	Specifies that the current drawing shall move and resize to maintain its row and column anchors (i.e. the object is anchored to the actual from and to row and column).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_EditAs) is located in §A.4.5. *end note*]

20.5.3.3 ST_RowID (Row ID)

This simple type specifies a row identification. The numerical value used for the row id should be non-negative and never exceed the number of total rows within the spreadsheet document.

This simple type's contents are a restriction of the W3C XML Schema int datatype.

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_RowID) is located in §A.4.5. *end note*]

21. DrawingML - Components Reference Material

The subordinate subclauses specify the semantics for the XML markup comprising DrawingML content, which can be used within the contents of WordprocessingML, SpreadsheetML, or PresentationML documents.

This portion of DrawingML describes components built on top of the previously defined framework to create charts, diagrams, etc..

21.1 DrawingML - Main

The following parts of the DrawingML Main namespace define additional base constructs for all kinds of DrawingML objects (e.g., paragraphs, text, tables, etc.).

21.1.1 Table of Contents

This subclause is informative.

End of informative text.

21.1.2 Paragraphs and Rich Formatting

The Paragraphs and Rich Formatting portion of the DrawingML framework stores text and related formatting information for a text body contained within a shape. Formatting for text within a shape can be broken down into three levels of precision, namely body, paragraph, and run formatting properties.

21.1.2.1 Body Formatting

Being the highest level of formatting available within a shape, the body properties allow for the manipulation of the text area as a whole. This means that all paragraphs and runs of text for the shape in question would be encompassed within here and, therefore, follow the text body style defined here.

[*Example*: Consider a shape that has three paragraphs within it, each with a different sized text. If this shape is resized to be smaller, then the text no longer fits the same way within the shape. Thus, we see that to maintain visual quality the size must be changed.

Rather than try to change each of the paragraphs to a smaller font size to fit in the shape, just use a body-level format property such as the normAutofit. This applies to all text within the shape and scale the text by a certain percentage in order to fit within the shape.

```
<p:txBody>
  <a:bodyPr>
    <a:normAutofit fontScale="20.000%" lnSpcReduction="20.000%"/>
  </a:bodyPr>
  …
  <a:p>
  …
  </a:p>
  <a:p>
  …
  </a:p>
  <a:p>
  …
  </a:p>
</p:txBody>
```

end example]

21.1.2.1.1 bodyPr (Body Properties)

This element defines the body properties for the text body within a shape.

[*Example*: Consider a shape with a text body that has some formatting properties associated with it. For the formatting of text body properties, the bodyPr element should be used as follows:

```
<p:sp>
  …
  <p:txBody>
    <a:bodyPr>
      (text body properties)
    </a:bodyPr>
  …
  </p:txBody>
</p:sp>
```

end example]

Attributes	Description

anchor (Anchor)	Specifies the anchoring position of the txBody within the shape. If this attribute is omitted, then a value of t, or top is implied.
	
	[*Example*: Consider the following DrawingML:
	
	`<p:txBody>`
	`  <a:bodyPr anchor="ctr" … />`
	`  …`
	`</p:txBody>`
	
	Here the text is vertically aligned in the center of the shape within which it is contained. *end example*]
	
	The possible values for this attribute are defined by the ST_TextAnchoringType simple type (§20.1.10.59).

anchorCtr (Anchor Center)	Specifies the centering of the text box. The way it works fundamentally is to determine the smallest possible "bounds box" for the text and then to center that "bounds box" accordingly. This is different than paragraph alignment, which aligns the text within the "bounds box" for the text. This flag is compatible with all of the different kinds of anchoring. If this attribute is omitted, then a value of `0` or `false` is implied.
	
	[*Example*: The text within this shape has been both vertically centered with the anchor attribute and horizontally centered with the anchorCtr attribute.
	
	`<p:txBody>`
	`  <a:bodyPr anchor="ctr" anchorCtr="1" … />`
	`  …`

Attributes	Description
	</p:txBody> *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
bIns (Bottom Inset)	Specifies the bottom inset of the bounding rectangle. Insets are used just as internal margins for text boxes within shapes. If this attribute is omitted, a value of 45720 or 0.05 inches is implied. [*Example*: Consider the following DrawingML: <p:txBody> <a:bodyPr lIns="91440" tIns="91440" rIns="91440" bIns="91440" … /> … </p:txBody> The text box having the above body properties has inset margins of 91440 or 0.1 inches on all four sides. *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
compatLnSpc (Compatible Line Spacing)	Specifies that the line spacing for this text body is decided in a simplistic manner using the font scene. If this attribute is omitted, a value of 0 or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
forceAA (Force Anti-Alias)	Forces the text to be rendered anti-aliased regardless of the font size. Certain fonts can appear grainy around their edges unless they are anti-aliased. Therefore this attribute allows for the specifying of which bodies of text should always be anti-aliased and which ones should not. If this attribute is omitted, then a value of 0 or falseis implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fromWordArt (From WordArt)	Specifies that text within this textbox is converted text from a WordArt object. This is more of a backwards compatibility attribute that is useful to the application from a tracking perspective. WordArt was the former way to apply text effects and therefore this attribute is useful in document conversion scenarios. If this attribute is omitted, then a value of 0 or falseis implied. [*Example*: Consider the following DrawingML: <p:txBody> <a:bodyPr wrap="none" fromWordArt="1" … /> … </p:txBody>

Attributes	Description
	end example] Because of the presence of the fromWordArt attribute the text within this shape can be mapped back to the corresponding WordArt during document conversion. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

horzOverflow (Text Horizontal Overflow)	Determines whether the text can flow out of the bounding box horizontally. This is used to determine what happens in the event that the text within a shape is too large for the bounding box it is contained within. If this attribute is omitted, then a value of overflow is implied.
	
	[*Example*: Consider the case where we have multiply paragraphs within a shape and the second is greater in length and causes text to flow outside the shape. By applying the clip value of the horzOverflow attribute as a body property this overflowing text now is cut off instead of extending beyond the bounds of the shape.
	
	`<p:txBody>`
	`  <a:bodyPr horzOverflow="clip" … />`
	`  …`
	`  <a:p>`
	`  …`
	`  (Some text)`
	`  …`
	`  </a:p>`
	`  <a:p>`
	`  …`
	`  (Some more text)`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TextHorzOverflowType simple type (§20.1.10.68).

lIns (Left Inset)	Specifies the left inset of the bounding rectangle. Insets are used just as internal margins for text boxes within shapes. If this attribute is omitted, then a value of 91440 or 0.1 inches is implied.
	
	[*Example*: Consider the following DrawingML:
	
	`<p:txBody>`
	`  <a:bodyPr lIns="91440" tIns="91440" rIns="91440"`
	`  bIns="91440" … />`
	`  …`
	`</p:txBody>`

Attributes	Description
	The text box having the above body properties has inset margins of 91440 or 0.1 inches on all four sides. *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).

numCol (Number of Columns)	Specifies the number of columns of text in the bounding rectangle. When applied to a text run this property takes the width of the bounding box for the text and divides it by the number of columns specified. These columns are then treated as overflow containers in that when the previous column has been filled with text the next column acts as the repository for additional text. When all columns have been filled and text still remains then the overflow properties set for this text body are used and the text is reflowed to make room for additional text. If this attribute is omitted, then a value of 1 is implied.
	
	[*Example*: Consider the case where a text area would need to be split up into four separate columns. Then simply specifying one paragraph with one run of text is enough to describe four columns of text here.
	
	`<p:txBody>`
	`  <a:bodyPr numCol="4" … />`
	`  <a:p>`
	`  <a:r>`
	`  …`
	`  (Some text)`
	`  …`
	`  </a:r>`
	`  </a:p>`
	`</p:txBody>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TextColumnCount simple type (§20.1.10.64).

rIns (Right Inset)	Specifies the right inset of the bounding rectangle. Insets are used just as internal margins for text boxes within shapes. If this attribute is omitted, then a value of 91440 or 0.1 inches is implied.
	
	[*Example*: Consider the following DrawingML:
	
	`<p:txBody>`
	`  <a:bodyPr lIns="91440" tIns="91440"`
	`  rIns="91440" bIns="91440" … />`
	`  …`
	`</p:txBody>`
	
	The text box having the above body properties has inset margins of 91440 or 0.1 inches

Attributes	Description
	on all four sides. *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).

rot (Rotation)	Specifies the rotation that is being applied to the text within the bounding box. If it not specified, the rotation of the accompanying shape is used. If it is specified, then this is applied independently from the shape. That is the shape can have a rotation applied in addition to the text itself having a rotation applied to it. If this attribute is omitted, then a value of 0, is implied.
	
	[*Example*: Consider the case where a shape has a rotation of 5400000, or 90 degrees clockwise applied to it. In addition to this, the text body itself has a rotation of -5400000, or 90 degrees counter-clockwise applied to it. Then the resulting shape would appear to be rotated but the text within it would appear as though it had not been rotated at all. The DrawingML specifying this would look like the following:
	
	`<p:sp>`
	`  <p:spPr>`
	`  <a:xfrm rot="5400000">`
	`  …`
	`  </a:xfrm>`
	`  </p:spPr>`
	`  …`
	`  <p:txBody>`
	`  <a:bodyPr rot="-5400000" … />`
	`  …`
	`  (Some text)`
	`  …`
	`  </p:txBody>`
	`</p:sp>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Angle simple type (§20.1.10.3).

rtlCol (Columns Right-To-Left)	Specifies whether columns are used in a right-to-left or left-to-right order. The usage of this attribute only sets the column order that is used to determine which column overflow text should go to next. If this attribute is omitted, then a value of `0` or `false`is implied in which case text starts in the leftmost column and flow to the right. [*Note*: This attribute in no way determines the direction of text but merely the direction in which multiple columns are used. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
spcCol (Space	Specifies the space between text columns in the text area. This should only apply when

Attributes	Description
Between Columns)	there is more than 1 column present. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_PositiveCoordinate32 simple type (§20.1.10.42).

spcFirstLastPara (Paragraph Spacing)	Specifies whether the before and after paragraph spacing defined by the user is to be respected. While the spacing between paragraphs is helpful, it is additionally useful to be able to set a flag as to whether this spacing is to be followed at the edges of the text body, in other words the first and last paragraphs in the text body. More precisely since this is a text body level property it should only effect the before paragraph spacing of the first paragraph and the after paragraph spacing of the last paragraph for a given text body. If this attribute is omitted, then a value of 0, or false is implied.
	
	[*Example*: Consider the case where spacing has been defined between multiple paragraphs within a text body using the spcBef and spcAft paragraph spacing attributes. For this text body however the user would like to not have this followed for the edge paragraphs and thus we have the following DrawingML.
	
	`<p:txBody>`
	`<a:bodyPr spcFirstLastPara="0" … />`
	`…`
	`<a:p>`
	`<a:pPr>`
	`<a:spcBef>`
	`<a:spcPts val="1800"/>`
	`</a:spcBef>`
	`<a:spcAft>`
	`<a:spcPts val="600"/>`
	`</a:spcAft>`
	`</a:pPr>`
	`…`
	`(Some text)`
	`…`
	`</a:p>`
	`<a:p>`
	`<a:pPr>`
	`<a:spcBef>`
	`<a:spcPts val="1800"/>`
	`</a:spcBef>`
	`<a:spcAft>`
	`<a:spcPts val="600"/>`
	`</a:spcAft>`
	`</a:pPr>`
	`…`
	`(Some text)`
	`…`
	`</a:p>`

Attributes	Description
	… </p:txBody> *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
tIns (Top Inset)	Specifies the top inset of the bounding rectangle. Insets are used just as internal margins for text boxes within shapes. If this attribute is omitted, then a value of 45720 or 0.05 inches is implied. [*Example*: Consider the following DrawingML: <p:txBody> <a:bodyPr lIns="91440" tIns="91440" rIns="91440" bIns="91440" … /> … </p:txBody> The text box having the above body properties has inset margins of 91440 or 0.1 inches on all four sides. *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
upright (Text Upright)	Specifies whether text should remain upright, regardless of the transform applied to it and the accompanying shape transform. If this attribute is omitted, then a value of 0, or false is implied. [*Example*: Consider text that has been rotated within the text body but has the upright flag set. <p:txBody> <a:bodyPr upright="1" rot="5400000" …/> … (Some text) … </p:txBody> This text appears as though no transform has been applied to it. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
vert (Vertical Text)	Determines if the text within the given text body should be displayed vertically. If this attribute is omitted, then a value of horz, or no vertical text is implied. [*Example*: Consider the case where the user needs to display text that appears vertical and has a right to left flow with respect to its columns.

Attributes	Description
	<p:txBody> <a:bodyPr vert="wordArtVertRtl" … /> … <a:p> … <a:t>This is</a:t> … </a:p> <a:p> … <a:t>some text.</a:t> … </a:p> </p:txBody> In the above sample DrawingML there are two paragraphs denoting a separation between the text otherwise which are known as either a line or paragraph break. Because wordArtVertRtl is used here this text is not only displayed in a stacked manner flowing from top to bottom but also have the first paragraph be displayed to the right of the second. This is because it is both vertical text and right to left. *end example*] The possible values for this attribute are defined by the ST_TextVerticalType simple type (§20.1.10.82).
vertOverflow (Text Vertical Overflow)	Determines whether the text can flow out of the bounding box vertically. This is used to determine what happens in the event that the text within a shape is too large for the bounding box it is contained within. If this attribute is omitted, then a value of overflow is implied. [*Example*: Consider the case where we have multiply paragraphs within a shape and the second causes text to flow outside the shape. By applying the clip value of the vertOverflow attribute as a body property this overflowing text is now cut off instead of extending beyond the bounds of the shape. <p:txBody> <a:bodyPr vertOverflow="clip" … /> … <a:p> … (Some text) … </a:p> <a:p> … (Some longer text) … </a:p>

Attributes	Description
	</p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextVertOverflowType simple type (§20.1.10.83).
wrap (Text Wrapping Type)	Specifies the wrapping options to be used for this text body. If this attribute is omitted, then a value of square is implied which wraps the text using the bounding text box. The possible values for this attribute are defined by the ST_TextWrappingType simple type (§20.1.10.84).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBodyProperties) is located in §A.4.1. *end note*]

21.1.2.1.2 noAutofit (No AutoFit)

This element specifies that text within the text body should not be auto-fit to the bounding box. Auto-fitting is when text within a text box is scaled in order to remain inside the text box. If this element is omitted, then noAutofit or auto-fit off is implied.

[*Example*: Consider a text box where the user wishes to have the text extend outside the bounding box. The following DrawingML would describe this.

```
<p:txBody>
  <a:bodyPr wrap="none" rtlCol="0">
    <a:noAutofit/>
  </a:bodyPr>
  <a:p>
    …
    <a:t>Some text</a:t>
    …
  </a:p>
</p:txBody>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextNoAutofit) is located in §A.4.1. *end note*]

21.1.2.1.3 normAutofit (Normal AutoFit)

This element specifies that text within the text body should be normally auto-fit to the bounding box. Auto-fitting is when text within a text box is scaled in order to remain inside the text box. If this element is omitted, then noAutofit or auto-fit off is implied.

[*Example*: Consider the situation where a user is building a diagram and needs to have the text for each shape that they are using stay within the bounds of the shape. An easy way this might be done is by using normAutofit. The following DrawingML illustrates how this might be accomplished.

```
<p:sp>
  <p:txBody>
    <a:bodyPr rtlCol="0" anchor="ctr">
      <a:normAutofit fontScale="92.000%" lnSpcReduction="20.000%"/>
    </a:bodyPr>
    …
    <a:p>
      …
      <a:t>Diagram Object 1</a:t>
      …
    </a:p>
  </p:txBody>
</p:sp>
<p:sp>
  <p:txBody>
    <a:bodyPr rtlCol="0" anchor="ctr">
      <a:normAutofit fontScale="92.000%" lnSpcReduction="20.000%"/>
    </a:bodyPr>
    …
    <a:p>
      …
      <a:t>Diagram Object 2</a:t>
      …
    </a:p>
  </p:txBody>
</p:sp>
```

In the above example there are two shapes that have normAutofit turned on so that when the user types more text within the shape that the text actually resizes to accommodate the new data. For the application to know how and to what degree the text should be resized two attributes are set for the auto-fit resize logic. *end example*]

Attributes	Description
fontScale (Font	Specifies the percentage of the original font size to which each run in the text body is

Attributes	Description
Scale)	scaled. In order to auto-fit text within a bounding box it is sometimes necessary to decrease the font size by a certain percentage. Using this attribute the font within a text box can be scaled based on the value provided. A value of 100% scales the text to 100%, while a value of 1% scales the text to 1%. If this attribute is omitted, then a value of 100% is implied. The possible values for this attribute are defined by the ST_TextFontScalePercentOrPercentString simple type (§20.1.10.66).
lnSpcReduction (Line Space Reduction)	Specifies the percentage amount by which the line spacing of each paragraph in the text body is reduced. The reduction is applied by subtracting it from the original line spacing value. Using this attribute the vertical spacing between the lines of text can be scaled by a percent amount. A value of 100% reduces the line spacing by 100%, while a value of 1% reduces the line spacing by one percent. If this attribute is omitted, then a value of 0% is implied. [*Note*: This attribute applies only to paragraphs with percentage line spacing. *end note*] The possible values for this attribute are defined by the ST_TextSpacingPercentOrPercentString simple type (§20.1.10.76).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextNormalAutofit) is located in §A.4.1. *end note*]

21.1.2.1.4 spAutoFit (Shape AutoFit)

This element specifies that a shape should be auto-fit to fully contain the text described within it. Auto-fitting is when text within a shape is scaled in order to contain all the text inside. If this element is omitted, then noAutofit or auto-fit off is implied.

[*Example*: Consider the situation where a user is building a diagram and needs to have the text for each shape that they are using stay within the bounds of the shape. An easy way this might be done is by using spAutofit. The following DrawingML illustrates how this might be accomplished.

```
<p:sp>
  <p:txBody>
    <a:bodyPr rtlCol="0" anchor="ctr">
      <a:spAutoFit/>
    </a:bodyPr>
    …
    <a:p>
      …
      <a:t>Diagram Object 1</a:t>
      …
    </a:p>
  </p:txBody>
</p:sp>
<p:sp>
  <p:txBody>
    <a:bodyPr rtlCol="0" anchor="ctr">
      <a:spAutoFit/>
    </a:bodyPr>
    …
    <a:p>
      …
      <a:t>Diagram Object 2</a:t>
      …
    </a:p>
  </p:txBody>
</p:sp>
```

In the above example there are two shapes that have spAutoFit turned on so that when the user types more text within the shape that the shape actually resizes to accommodate the new data. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextShapeAutofit) is located in §A.4.1. *end note*]

21.1.2.2 Paragraph Formatting

This level of formatting allows for more granular control of text within a shape. Properties here apply to all text residing within the corresponding paragraph. This intermediate property level allows freedom to assign what would seem like lower level properties to a larger group of text. Along with this the paragraph property level also allows what would seem like larger group properties to a more granular set of text. This makes for a property level that is quite versatile in its ability to define formatting on text within a shape.

[*Example*: For instance consider the case where a paragraph of text would need to have bullets applied to it. At first one might think that this formatting must be done at the text run level as it can seem run specific. Much to the contrary this is a paragraph level property and is applied to multiple runs of text. As an example we have once again our three paragraphs with the second having bullets applied to it.


```
<a:p>
…
</a:p>
<a:p>
  <a:pPr>
    <a:buFont typeface="Wingdings"/>
    <a:buChar typeface="ü"/>
  <a:pPr>
  <a:r>
    <a:rPr lang="en-US" dirty="0" smtClean="0"/>
    <a:t>This Paragraph of Text Will Have a Bullet.</a:t>
  </a:r>
</a:p>
<a:p>
…
</a:p>
```

Here we see that the paragraph is formatted to have character bullets for each new line of text that is encountered. In particular this paragraph has the "ü" character applied which in the "Wingdings" font is the checkmark character. The other paragraphs are not effected by this paragraph's bullet formatting and should have their text remain unformatted.

end example]

21.1.2.2.1 br (Text Line Break)

This element specifies the existence of a vertical line break between two runs of text within a paragraph. In addition to specifying a vertical space between two runs of text, this element can also have run properties specified via the rPr child element. This sets the formatting of text for the line break so that if text is later inserted there that a new run can be generated with the correct formatting.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:r>
      …
      <a:t>Text Run 1.</a:t>
      …
    </a:r>
```

```
      <a:br/>
      <a:r>
        …
        <a:t>Text Run 2.</a:t>
        …
      </a:r>
    </a:p>
  </p:txBody>
```

This paragraph has two runs of text laid out in a vertical fashion with a line break in between them. This line break acts much like a carriage return would within a normal run of text. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextLineBreak) is located in §A.4.1. *end note*]

21.1.2.2.2 defPPr (Default Paragraph Style)

This element specifies the paragraph properties that are to be applied when no other paragraph properties have been specified. If this attribute is omitted, then it is left to the application to decide the set of default paragraph properties that should be applied.

[*Example*: Consider the DrawingML shown below.

```
  <p:txBody>
    …
    <a:lstStyle>
      <a:defPPr>
        <a:buNone/>
      </a:defPPr>
    </a:lstStyle>
    <a:p>
      …
      <a:t>Sample Text</a:t>
      …
    </a:p>
  </p:txBody>
```

The above paragraph follows the properties described in defPPr if no overriding properties are specified within the pPr element. *end example*]

Attributes	Description
algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied.

Attributes	Description
	Sample text Sample text

[*Example*: Consider the case where the user wishes to have two columns of text that have a justified alignment, much like text within a book. The following DrawingML could describe this.

```
<p:txBody>
  <a:bodyPr numCol="2" spcCol="914400"…/>
    <a:normAutofit/>
  </a:bodyPr>
  …
  <a:p>
    <a:pPr marL="0" algn="just">
      <a:buNone/>
    </a:pPr>
    …
    <a:t>Sample Text …</a:t>
    …
  </a:p>
</p:txBody>
```

end example]

The possible values for this attribute are defined by the ST_TextAlignType simple type (§20.1.10.58).

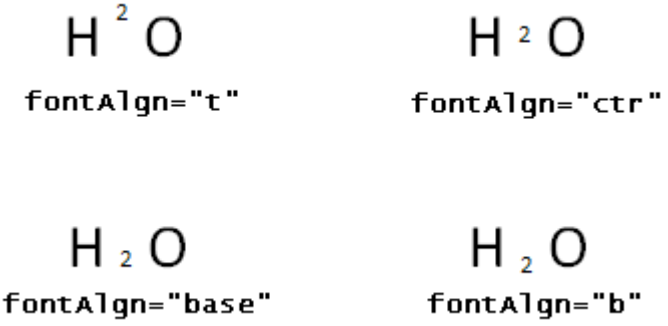
Attributes	Description
defTabSz (Default Tab Size)	Specifies the default size for a tab character within this paragraph. This attribute should be used to describe the spacing of tabs within the paragraph instead of a leading indentation tab. For indentation tabs there are the marL and indent attributes to assist with this.

[*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this.

```
<p:txBody>
  …
```


Attributes	Description
	<a:p> <a:pPr defTabSz="376300" …/> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows. <p:txBody> … <a:p> <a:pPr eaLnBrk="0" …/> … <a:t>Sample text (Long word)</a:t> … </a:p> </p:txBody> Sample text Sample text Sample text supercalifr agilisticexpialidocious Sample text Sample text Sample text supercalifragilisticexpialidocious *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

Attributes	Description

fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied.
	
	[*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below.
	
	 
	
	`<a:txtBody>`
	`…`
	`<a:pPr fontAlgn="b" …/>`
	`…`
	`<a:r>`
	`<a:rPr …/>`
	`<a:t>H </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr sz="1200" …/>`
	`<a:t>2</a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr …/>`
	`<a:t>O</a:t>`
	`</a:r>`
	`…`
	`</p:txBody>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.65).

Attributes	Description
hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this attribute is omitted, then a value of -342900 is implied.
	
	Here is some text Sample text
	
	[*Example*: Consider the scenario where the user now wanted to add a paragraph indentation to the first line of text in their two column format book.
	
	`<p:txBody>`
	`<a:bodyPr numCol="2" spcCol="914400"…/>`
	`<a:normAutofit/>`
	`</a:bodyPr>`
	`…`
	`<a:p>`
	`<a:pPr marL="0" indent="571500" algn="just">`
	`<a:buNone/>`
	`</a:pPr>`
	`…`
	`<a:t>Here is some…</a:t>`
	`…`
	`</a:p>`
	`</p:txBody>`
	
	By adding the indent attribute the user has effectively added a first line indent to this paragraph of text. *end example*]

Attributes	Description
	The possible values for this attribute are defined by the ST_TextIndent simple type (§20.1.10.69).

latinLnBrk (Latin Line Break)	Specifies whether a Latin word can be broken in half and wrapped onto the next line without a hyphen being added. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. It is not present within the existence of normal breakable Latin words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied.
	
	[*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr latinLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values. [*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.

Attributes	Description
	<p:txBody> … <a:p> <a:pPr lvl="1" …/> … <a:t>Sample text</a:t> … </a:p> </p:txBody> *end example*] [*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*] The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.70).
marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).
rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this attribute is omitted, then a value of 0, or left-to-right is implied. [*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.

Attributes	Description

	
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`
	`<a:t>تجربة</a:t>`
	`</a:r>`
	`</a:p>`
	`<a:p>`
	`<a:pPr rtl="1"/>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="0"/>`
	`</a:rPr>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:t>تجربة </a:t>`
	`</a:r>`
	`</a:p>`
	`</p:txBody>`
	
	end example]
	
	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_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.2.3 endParaRPr (End Paragraph Run Properties)

This element specifies the text run properties that are to be used if another run is inserted after the last run specified. This effectively saves the run property state so that it can be applied when the user enters additional text. If this element is omitted, then the application can determine which default properties to apply. It is

recommended that this element be specified at the end of the list of text runs within the paragraph so that an orderly list is maintained.

Attributes	Description
altLang (Alternative Language)	Specifies the alternate language to use when the generating application is displaying the user interface controls. If this attribute is omitted, than the lang attribute is used here. The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).

b (Bold)	Specifies whether a run of text is formatted as bold text. If this attribute is omitted, than a value of 0, or false is assumed.
	[*Example*: Consider the DrawingML shown below.
	`<a:p>`
	`…`
	`<a:rPr b="1"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	The above run of text is formatted as bold text. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

baseline (Baseline)	Specifies the baseline for both the superscript and subscript fonts. The size is specified using a percentage where 1% is equal to 1 percent of the font size and 100% is equal to 100 percent font of the font size. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
bmk (Bookmark Link Target)	Specifies the link target name that is used to reference to the proper link properties in a custom XML part within the document. The possible values for this attribute are defined by the W3C XML Schema string datatype.
cap (Capitalization)	Specifies the capitalization that is to be applied to the text run. This is a render-only modification and does not affect the actual characters stored in the text run. This attribute is also distinct from the toggle function where the actual characters stored in the text run are changed. The possible values for this attribute are defined by the ST_TextCapsType simple type (§20.1.10.63).
dirty (Dirty)	Specifies that the content of a text run has changed since the proofing tools have last been run. Effectively this flags text that is to be checked again by the generating application for mistakes such as spelling, grammar, etc. The possible values for this attribute are defined by the W3C XML Schema boolean

Attributes	Description
	datatype.
err (Spelling Error)	Specifies that when this run of text was checked for spelling, grammar, etc. that a mistake was indeed found. This allows the generating application to effectively save the state of the mistakes within the document instead of having to perform a full pass check upon opening the document. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

i (Italics)	Specifies whether a run of text is formatted as italic text. If this attribute is omitted, than a value of 0, or false is assumed.
	
	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`…`
	`<a:rPr i="1"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	
	The above run of text is formatted as italic text. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

kern (Kerning)	Specifies the minimum font size at which character kerning occurs for this text run. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. If this attribute is omitted, than kerning occurs for all font sizes down to a 0 point font. The possible values for this attribute are defined by the ST_TextNonNegativePoint simple type (§20.1.10.72).
kumimoji (Kumimoji)	Specifies whether the numbers contained within vertical text continue vertically with the text or whether they are to be displayed horizontally while the surrounding characters continue in a vertical fashion. If this attribute is omitted, than a value of 0, or false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
lang (Language ID)	Specifies the language to be used when the generating application is displaying the user interface controls. If this attribute is omitted, than the generating application can select a language of its choice. The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).

Attributes	Description
noProof (No Proofing)	Specifies that a run of text has been selected by the user to not be checked for mistakes. Therefore if there are spelling, grammar, etc mistakes within this text the generating application should ignore them. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
normalizeH (Normalize Heights)	Specifies the normalization of height that is to be applied to the text run. This is a render-only modification and does not affect the actual characters stored in the text run. This attribute is also distinct from the toggle function where the actual characters stored in the text run are changed. If this attribute is omitted, than a value of 0, or false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
smtClean (SmartTag Clean)	Specifies whether or not a text run has been checked for smart tags. This attribute acts much like the dirty attribute dose for the checking of spelling, grammar, etc. A value of true here indicates to the generating application that this text run should be checked for smart tags. If this attribute is omitted, than a value of 0, or false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

smtId (SmartTag ID)	Specifies a smart tag identifier for a run of text. This ID is unique throughout the presentation and is used to reference corresponding auxiliary information about the smart tag. [*Note*: For a complete definition of smart tags, which are semantically identical throughout Office Open XML, see §17.5.1. *end note*]
	
	[*Example*: Consider the following DrawingML:
	
	`<p:txBody>`
	`  <a:bodyPr/>`
	`  <a:lstStyle/>`
	`  <a:p>`
	`  <a:r>`
	`  <a:rPr lang="en-US" dirty="0" smtId="1"/>`
	`  <a:t>CNTS</a:t>`
	`  </a:r>`
	`  <a:endParaRPr lang="en-US" dirty="0"/>`
	`  </a:p>`
	`</p:txBody>`
	
	The text run has a smtId attribute value of 1, which denotes that the text should be inspected for smart tag information, which in this case maps to a stock ticker symbol. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema unsignedInt

Attributes	Description
	datatype.
spc (Spacing)	Specifies the spacing between characters within a text run. This spacing is specified numerically and should be consistently applied across the entire run of text by the generating application. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. If this attribute is omitted than a value of 0 or no adjustment is assumed. The possible values for this attribute are defined by the ST_TextPoint simple type (§20.1.10.73).

strike (Strikethrough)	Specifies whether a run of text is formatted as strikethrough text. If this attribute is omitted, than no strikethrough is assumed.
	
	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`…`
	`<a:rPr strike="sngStrike"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	
	The above run of text is formatted as single strikethrough text. *end example*]
	
	The possible values for this attribute are defined by the ST_TextStrikeType simple type (§20.1.10.78).

sz (Font Size)	Specifies the size of text within a text run. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. If this attribute is omitted, than the value in defRPr should be used.
	
	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`…`
	`<a:rPr sz="1200"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	
	The above run of text is formatted with a 12 point text size. *end example*]
	
	The possible values for this attribute are defined by the ST_TextFontSize simple type (§20.1.10.67).

Attributes	Description

u (Underline)	Specifies whether a run of text is formatted as underlined text. If this attribute is omitted, than no underline is assumed.
	
	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`…`
	`<a:rPr u="sng"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	
	The above run of text is formatted as single underline text. *end example*]
	
	The possible values for this attribute are defined by the ST_TextUnderlineType simple type (§20.1.10.81).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextCharacterProperties) is located in §A.4.1. *end note*]

21.1.2.2.4 fld (Text Field)

This element specifies a text field which contains generated text that the application should update periodically. Each piece of text when it is generated is given a unique identification number that is used to refer to a specific field. At the time of creation the text field indicates the kind of text that should be used to update this field. This update type is used so that all applications that did not create this text field can still know what kind of text it should be updated with. Thus the new application can then attach an update type to the text field id for continual updating.

[*Example*: Consider a slide within a presentation that needs to have the slide number placed on the slide. The following DrawingML can be used to describe such a situation.

```
<p:txBody>
  <a:bodyPr/>
    <a:lstStyle/>
    <a:p>
      <a:fld id="{424CEEAC-8F67-4238-9622-1B74DC6E8318}" type="slidenum">
      <a:rPr lang="en-US" smtClean="0"/>
      <a:pPr/>
      <a:t>3</a:t>
    </a:fld>
```



```
      <a:endParaRPr lang="en-US"/>
    </a:p>
  </p:txBody>
```

end example]

Attributes	Description
id (Field ID)	Specifies the unique to this document, host specified token that is used to identify the field. This token is generated when the text field is created and persists in the file as the same token until the text field is removed. Any application should check the document for conflicting tokens before assigning a new token to a text field. The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).
type (Field Type)	Specifies the type of text that should be used to update this text field. This is used to inform the rendering application what text it should use to update this text field. There are no specific syntax restrictions placed on this attribute. The generating application can use it to represent any text that should be updated before rendering the presentation. Reserved Values:

Value	Description
slidenum	presentation slide number
datetime	default date time format for the rendering application
datetime1	MM/DD/YYYY date time format [*Example:* 10/12/2007 *end example*]
datetime2	Day, Month DD, YYYY date time format [*Example:* Friday, October 12, 2007 *end example*]
datetime3	DD Month YYYY date time format [*Example:* 12 October 2007 *end example*]
datetime4	Month DD, YYYY date time format [*Example:* October 12, 2007 *end example*]
datetime5	DD-Mon-YY date time format [*Example:* 12-Oct-07 *end example*]
datetime6	Month YY date time format [*Example:* October 07 *end example*]
datetime7	Mon-YY date time format [*Example:* Oct-07 *end example*]
datetime8	MM/DD/YYYY hh:mm AM/PM date time format [*Example:* 10/12/2007 4:28 PM *end example*]
datetime9	MM/DD/YYYY hh:mm:ss AM/PM date time format [*Example:* 10/12/2007 4:28:34 PM *end example*]
datetime10	hh:mm date time format [*Example:* 16:28 *end example*]

Attributes	Description
	`datetime11` — hh:mm:ss date time format [*Example:* 16:28:34 *end example*] `datetime12` — hh:mm AM/PM date time format [*Example:* 4:28 PM *end example*] `datetime13` — hh:mm:ss: AM/PM date time format [*Example:* 4:28:34 PM *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_TextField) is located in §A.4.1. *end note*]

21.1.2.2.5 lnSpc (Line Spacing)

This element specifies the vertical line spacing that is to be used within a paragraph. This can be specified in two different ways, percentage spacing and font point spacing. If this element is omitted then the spacing between two lines of text should be determined by the point size of the largest piece of text within a line.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  <a:p>
    <a:pPr>
      <a:lnSpc>
        <a:spcPct val="200%"/>
      </a:lnSpc>
    </a:pPr>
    <a:r>
      <a:rPr lang="en-US" dirty="0" smtClean="0"/>
      <a:t>Some</a:t>
    </a:r>
    <a:br>
      <a:rPr lang="en-US" smtClean="0"/>
    </a:br>
    <a:r>
      <a:rPr lang="en-US" dirty="0" smtClean="0"/>
      <a:t>Text</a:t>
    </a:r>
  </a:p>
</p:txBody>
```


This paragraph has two lines of text that have percentage based vertical spacing. This kind of spacing should change based on the size of the text involved as its size is calculated as a percentage of this. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextSpacing) is located in §A.4.1. *end note*]

21.1.2.2.6 p (Text Paragraphs)

This element specifies the presence of a paragraph of text within the containing text body. The paragraph is the highest level text separation mechanism within a text body. A paragraph can contain text paragraph properties associated with the paragraph. If no properties are listed then properties specified in the defPPr element are used.

[*Example*: Consider the case where the user would like to describe a text body that contains two paragraphs. The requirement for these paragraphs is that one be right aligned and the other left aligned. The following DrawingML would specify a text body such as this.

```
<p:txBody>
    …
    <a:p>
    <a:pPr algn="r">
    </a:pPr>
    …
    <a:t>Some text</a:t>
    …
  </a:p>
  <a:p>
    <a:pPr algn="l">
    </a:pPr>
    …
    <a:t>Some text</a:t>
    …
  </a:p>
</p:txBody>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextParagraph) is located in §A.4.1. *end note*]

21.1.2.2.7 pPr (Text Paragraph Properties)

This element contains all paragraph level text properties for the containing paragraph. These paragraph properties should override any and all conflicting properties that are associated with the paragraph in question.

[*Example*: Consider the DrawingML shown below.


```
<a:p>
  <a:pPr marL="0" algn="ctr">
    <a:buNone/>
  </a:pPr>
  …
  <a:t>Some Text</a:t>
  …
</a:p>
```

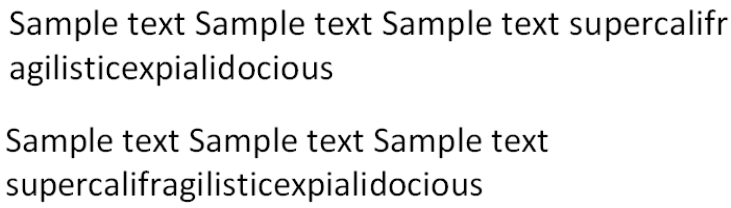
The paragraph described above is formatting with a left margin of 0 and has all of text runs contained within it centered about the horizontal median of the bounding box for the text body. *end example*]

[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]

Attributes	Description
algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied. Sample text [*Example*: Consider the case where the user wishes to have two columns of text that have a justified alignment, much like text within a book. The following DrawingML could describe this. `<p:txBody>` `  <a:bodyPr numCol="2" spcCol="914400"…/>` `  <a:normAutofit/>` `  </a:bodyPr>` `  …` `  <a:p>`

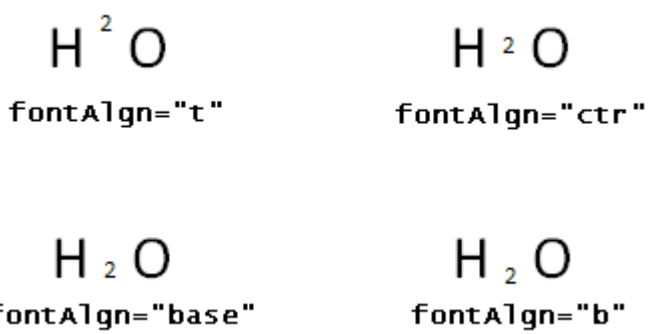
Attributes	Description
	<a:pPr marL="0" algn="just"> <a:buNone/> </a:pPr> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextAlignType simple type (§20.1.10.58).
defTabSz (Default Tab Size)	Specifies the default size for a tab character within this paragraph. This attribute should be used to describe the spacing of tabs within the paragraph instead of a leading indentation tab. For indentation tabs there are the marL and indent attributes to assist with this. [*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this. <p:txBody> … <a:p> <a:pPr defTabSz="376300" …/> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken

Attributes	Description

	for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr eaLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied. [*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below.

Attributes	Description

	
	[*example*:
	`<a:txtBody>`
	`…`
	`<a:pPr fontAlgn="b" …/>`
	`…`
	`<a:r>`
	`<a:rPr …/>`
	`<a:t>H </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr sz="1200" …/>`
	`<a:t>2</a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr …/>`
	`<a:t>O</a:t>`
	`</a:r>`
	`…`
	`</p:txBody>`
	end example]
	
	The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.65).

hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this

Attributes	Description

	attribute is omitted, then a value of -342900 is implied.
	
	Here is some Sample text
	
	[*Example*: Consider the scenario where the user now wanted to add a paragraph indentation to the first line of text in their two column format book.
	
	<p:txBody>
	  <a:bodyPr numCol="2" spcCol="914400"…/>
	    <a:normAutofit/>
	    </a:bodyPr>
	  …
	  <a:p>
	    <a:pPr marL="0" indent="571500" algn="just">
	      <a:buNone/>
	    </a:pPr>
	    …
	    <a:t>Here is some…</a:t>
	    …
	  </a:p>
	</p:txBody>
	
	By adding the indent attribute the user has effectively added a first line indent to this paragraph of text. *end example*]
	
	The possible values for this attribute are defined by the ST_TextIndent simple type (§20.1.10.69).

latinLnBrk (Latin Line Break)	Specifies whether a Latin word can be broken in half and wrapped onto the next line without a hyphen being added. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. It is not present within the existence of normal breakable Latin words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not

Attributes	Description

	be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr latinLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values.
	
	[*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr lvl="1" …/>`
	`  …`
	`  <a:t>Sample text</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	end example]

Attributes	Description
	[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*] The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.70).
marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).

rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this attribute is omitted, then a value of 0, or left-to-right is implied.
	
	[*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.
	
	 
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`

Attributes	Description
	<a:t>تجربة_</a:t> </a:r> </a:p> <a:p> <a:pPr rtl="1"/> <a:r> <a:rPr> <a:rtl w:val="0"/> </a:rPr> <a:t>Test </a:t> </a:r> <a:r> <a:t>تجربة</a:t> </a:r> </a:p> </p:txBody> *end example*] 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_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.2.8 rtl (Right to Left Run)

This element specifies whether the contents of this run shall have right-to-left characteristics. Specifically, the following behaviors are applied when this element's val attribute is `true` (or an equivalent):

- Formatting – When the contents of this run are displayed, all characters shall be treated as complex script characters. This means that the values of the cs element (§21.1.2.3.1) shall be used to determine the font face.
- Character Directionality Override – When the contents of this run are displayed, this property acts as a right-to-left override for characters which are classified as follows (using the Unicode Character Database):
  - Weak types except European Number, European Number Terminator, Common Number Separator, Arabic Number and (for Hebrew text) European Number Separator when constituting part of a number
  - Neutral types
- [*Rationale*: This override allows applications to store and utilize higher-level information beyond that implicitly derived from the Unicode Bidirectional algorithm. For example, if the string "first second" appears in a right-to-left paragraph inside a document, the Unicode algorithm would always result in

"first second" at display time (since the neutral character is surrounded by strongly classified characters). However, if the whitespace was entered using a right-to-left input method (e.g. a Hebrew keyboard), then that character could be classified as RTL using this property, allowing the display of "second first" in a right-to-left paragraph, since the user explicitly asked for the space in a right-to-left context. *end rationale*]

This element provides information used to resolve the (Unicode) classifications of individual characters as either L, R, AN or EN. Once this is determined, the line should be displayed subject to the recommendation of the Unicode Bidirectional Algorithm in reordering resolved levels.

This property shall not be used with strong left-to-right text. Any behavior under that condition is unspecified. This property, when off, should not be used with strong right-to-left text. Any behavior under that condition is unspecified.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then right to left characteristics shall not be applied to the contents of this run.

[*Example*: Consider the following DrawingML visual content: "`first second,` أولى ثاني". This content might appear as follows within its parent paragraph:

```
<a:p>
  <a:r>
    <a:t>first second, </w:t>
  </a:r>
  <a:r>
    <a:rPr>
      <a:rtl/>
    </a:rPr>
    <a:t>أولى</a:t>
  </a:r>
  <a:r>
    <a:rPr>
      <a:rtl/>
    </a:rPr>
    <a:t> </a:t>
  </a:r>
  <a:r>
    <a:rPr>
      <a:rtl/>
    </a:rPr>
    <a:t>ثاني</a:t>
  </a:r>
</a:p>
```


The presence of the rtl element on the second, third, and fourth runs specifies that:

- The formatting on those runs is specified using the complex-script property variants.
- The whitespace character is treated as right-to-left.

Note that the second, third and fourth runs could be joined as one run with the rtl element specified.

end example]

Attributes	Description
val (On/Off Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0` or `false` specifies that the property shall be explicitly turned off. [*Example*: For example, consider the following on/off property: `<… val="false"/>` The val attribute explicitly declares that the property is `false`. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.4.1. *end note*]

21.1.2.2.9 spcAft (Space After)

This element specifies the amount of vertical white space that is present after a paragraph. This space is specified in either percentage or points via the child elements spcPct and spcPts.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:spcBef>
        <a:spcPts val="1800"/>
      </a:spcBef>
      <a:spcAft>
        <a:spcPts val="600"/>
      </a:spcAft>
    </a:pPr>
```

```
      …
      <a:t>Sample Text</a:t>
      …
    </a:p>
    …
  </p:txBody>
```

The above paragraph of text is formatted to have a spacing both before and after the paragraph text. The spacing before is a size of 18 points, or value=1800 and the spacing after is a size of 6 points, or value=600. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextSpacing) is located in §A.4.1. *end note*]

21.1.2.2.10 spcBef (Space Before)

This element specifies the amount of vertical white space that is present before a paragraph. This space is specified in either percentage or points via the child elements spcPct and spcPts.

[*Example*: Consider the DrawingML shown below.

```
  <p:txBody>
    …
    <a:p>
      <a:pPr …>
        <a:spcBef>
          <a:spcPts val="1800"/>
        </a:spcBef>
        <a:spcAft>
          <a:spcPts val="600"/>
        </a:spcAft>
      </a:pPr>
      …
      <a:t>Sample Text</a:t>
      …
    </a:p>
    …
  </p:txBody>
```

The above paragraph of text is formatted to have a spacing both before and after the paragraph text. The spacing before is a size of 18 points, or value=1800 and the spacing after is a size of 6 points, or value=600. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextSpacing) is located in §A.4.1. *end note*]

21.1.2.2.11 spcPct (Spacing Percent)

This element specifies the amount of white space that is to be used between lines and paragraphs in the form of a percentage of the text size. The text size that is used to calculate the spacing here is the text for each run, with the largest text size having precedence. That is if there is a run of text with 10 point font and within the same paragraph on the same line there is a run of text with a 12 point font size then the 12 point should be used to calculate the spacing to be used.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:spcBef>
        <a:spcPct val="200%"/>
      </a:spcBef>
    </a:pPr>
    …
    <a:t>Sample Text</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The above paragraph of text is formatted to have a spacing before the paragraph text. This spacing is 200% of the size of the largest text on each line. *end example*]

Attributes	Description
val (Value)	Specifies the percentage of the size that the white space should be. The possible values for this attribute are defined by the ST_TextSpacingPercentOrPercentString simple type (§20.1.10.76).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextSpacingPercent) is located in §A.4.1. *end note*]

21.1.2.2.12 spcPts (Spacing Points)

This element specifies the amount of white space that is to be used between lines and paragraphs in the form of a text point size. The size is specified using points where 100 is equal to 1 point font and 1200 is equal to 12 point.

[*Example*: Consider the DrawingML shown below.


```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:spcBef>
        <a:spcPts val="1400"/>
      </a:spcBef>
    </a:pPr>
    …
    <a:t>Sample Text</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The above paragraph of text is formatted to have a spacing before the paragraph text. This spacing is a size of 14 points due to val="1400". *end example*]

Attributes	Description
val (Value)	Specifies the size of the white space in point size. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. The possible values for this attribute are defined by the ST_TextSpacingPoint simple type (§20.1.10.77).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextSpacingPoint) is located in §A.4.1. *end note*]

21.1.2.2.13 tab (Tab Stop)

This element specifies a single tab stop to be used on a line of text when there are one or more tab characters present within the text. When there is more than one present than they should be utilized in increasing position order which is specified via the pos attribute.

[*Example*: Consider the DrawingML shown below.


```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:tabLst>
        <a:tab pos="2292350" algn="l"/>
        <a:tab pos="2627313" algn="l"/>
        <a:tab pos="2743200" algn="l"/>
        <a:tab pos="2974975" algn="l"/>
      </a:tabLst>
    </a:pPr>
    …
    <a:t>Sample Text</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The paragraph within which this <a:tab> information resides has a total of 4 unique tab stops that should be listed in order of increasing position. Along with specifying the tab position each tab allows for the specifying of an alignment. *end example*]

Attributes	Description
algn (Tab Alignment)	Specifies the alignment that is to be applied to text using this tab stop. If this attribute is omitted then the application default for the generating application. The possible values for this attribute are defined by the ST_TextTabAlignType simple type (§20.1.10.79).
pos (Tab Position)	Specifies the position of the tab stop relative to the left margin. If this attribute is omitted then the application default for tab stops is used. The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextTabStop) is located in §A.4.1. *end note*]

21.1.2.2.14 tabLst (Tab List)

This element specifies the list of all tab stops that are to be used within a paragraph. These tabs should be used when describing any custom tab stops within the document. If these are not specified then the default tab stops of the generating application should be used.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextTabStopList) is located in §A.4.1. *end note*]

21.1.2.3 Run Formatting

Run level formatting is the most granular property level and allows for the specifying of all low level text properties. The text run is what all paragraphs are derived from and thus specifying various properties per run allows for a diversely formatted text paragraph.

[*Example*: Consider the case where have multiple runs within a paragraph and you wish to apply bold to only one of them without having to split up the text into higher level XML groups. To do this we would simply apply the bold run property to the text run that we wish to format as shown below.

```
<a:r>
…
</a:r>
<a:r>
  <a:rPr lang="en-US" b="1" dirty="0" smtClean="0"/>
  <a:t>This text will be bold</a:t>
</a:r>
<a:r>
…
</a:r>
```

end example]

21.1.2.3.1 cs (Complex Script Font)

This element specifies that a complex script font be used for a specific run of text. This font is specified with a typeface attribute much like the others but is specifically classified as a complex script font.

[*Example*: Consider the DrawingML shown below.

```
<a:r>
  <a:rPr …>
    <a:cs typeface="Sample Font"/>
  </a:rPr>
  <a:t>Sample Text</a:t>
</a:r>
```

The above run of text is rendered using the complex script font "Sample Font". *end example*]

If the specified font is not available on a system being used for rendering, then the attributes of this element can be utilized to select an alternative font.

Attributes	Description
charset (Similar	Specifies the character set which is supported by the parent font. This information can be

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>Character Set)</td><td>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:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x00</td><td>Specifies the ANSI character set. (IANA name iso-8859-1)</td></tr>
<tr><td>0x01</td><td>Specifies the default character set.</td></tr>
<tr><td>0x02</td><td>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.</td></tr>
<tr><td>0x4D</td><td>Specifies a Macintosh (Standard Roman) character set. (IANA name macintosh)</td></tr>
<tr><td>0x80</td><td>Specifies the JIS character set. (IANA name shift_jis)</td></tr>
<tr><td>0x81</td><td>Specifies the Hangul character set. (IANA name ks_c_5601-1987)</td></tr>
<tr><td>0x82</td><td>Specifies a Johab character set. (IANA name KS C-5601-1992)</td></tr>
<tr><td>0x86</td><td>Specifies the GB-2312 character set. (IANA name GBK)</td></tr>
<tr><td>0x88</td><td>Specifies the Chinese Big Five character set. (IANA name Big5)</td></tr>
<tr><td>0xA1</td><td>Specifies a Greek character set. (IANA name windows-1253)</td></tr>
<tr><td>0xA2</td><td>Specifies a Turkish character set. (IANA name iso-8859-9)</td></tr>
<tr><td>0xA3</td><td>Specifies a Vietnamese character set. (IANA name windows-1258)</td></tr>
<tr><td>0xB1</td><td>Specifies a Hebrew character set. (IANA name windows-1255)</td></tr>
<tr><td>0xB2</td><td>Specifies an Arabic character set. (IANA name windows-1256)</td></tr>
<tr><td>0xBA</td><td>Specifies a Baltic character set. (IANA name windows-1257)</td></tr>
<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)</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.</td></tr>
</table>

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>The possible values for this attribute are defined by the ST_Panose simple type (§22.9.2.8).</td></tr>
<tr><td>pitchFamily
(Similar Font Family)</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 an octet value 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>
<tr><td>0x52</td><td>VARIABLE PITCH + DECORATIVE FONT FAMILY</td></tr>
</table>

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.

[Note: Although the attribute name is pitchFamily, the integer value of this attribute specifies the font family with higher 4 bits and the font pitch with lower 4 bits. end note]

The possible values for this attribute are defined by the W3C XML Schema byte datatype.</td></tr>
<tr><td>typeface (Text Typeface)</td><td>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</td></tr>
</table>

Attributes	Description
	should be utilized in order to select an alternate font. The possible values for this attribute are defined by the ST_TextTypeface simple type (§20.1.10.80).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextFont) is located in §A.4.1. *end note*]

21.1.2.3.2 defRPr (Default Text Run Properties)

This element contains all default run level text properties for the text runs within a containing paragraph. These properties are to be used when overriding properties have not been defined within the rPr element.

[*Example*: Consider the DrawingML shown below.

```
<a:p>
  …
  <a:rPr u="sng"/>
  …
  <a:t>Some Text</a:t>
  …
</a:p>
```

The run of text described above is formatting with a single underline of text matching color. *end example*]

Attributes	Description
altLang (Alternative Language)	Specifies the alternate language to use when the generating application is displaying the user interface controls. If this attribute is omitted, than the lang attribute is used here. The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).
b (Bold)	Specifies whether a run of text is formatted as bold text. If this attribute is omitted, than a value of 0, or false is assumed. [*Example*: Consider the DrawingML shown below. <a:p> … <a:rPr b="1"/> … <a:t>Some Text</a:t> … </a:p> The above run of text is formatted as bold text. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

Attributes	Description
baseline (Baseline)	Specifies the baseline for both the superscript and subscript fonts. The size is specified using a percentage where 1% is equal to 1 percent of the font size and 100% is equal to 100 percent font of the font size. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
bmk (Bookmark Link Target)	Specifies the link target name that is used to reference to the proper link properties in a custom XML part within the document. The possible values for this attribute are defined by the W3C XML Schema string datatype.
cap (Capitalization)	Specifies the capitalization that is to be applied to the text run. This is a render-only modification and does not affect the actual characters stored in the text run. This attribute is also distinct from the toggle function where the actual characters stored in the text run are changed. The possible values for this attribute are defined by the ST_TextCapsType simple type (§20.1.10.63).
dirty (Dirty)	Specifies that the content of a text run has changed since the proofing tools have last been run. Effectively this flags text that is to be checked again by the generating application for mistakes such as spelling, grammar, etc. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
err (Spelling Error)	Specifies that when this run of text was checked for spelling, grammar, etc. that a mistake was indeed found. This allows the generating application to effectively save the state of the mistakes within the document instead of having to perform a full pass check upon opening the document. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

i (Italics)	Specifies whether a run of text is formatted as italic text. If this attribute is omitted, than a value of 0, or false is assumed.
	
	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`…`
	`<a:rPr i="1"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	
	The above run of text is formatted as italic text. *end example*]

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
kern (Kerning)	Specifies the minimum font size at which character kerning occurs for this text run. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. If this attribute is omitted, than kerning occurs for all font sizes down to a 0 point font. The possible values for this attribute are defined by the ST_TextNonNegativePoint simple type (§20.1.10.72).
kumimoji (Kumimoji)	Specifies whether the numbers contained within vertical text continue vertically with the text or whether they are to be displayed horizontally while the surrounding characters continue in a vertical fashion. If this attribute is omitted, than a value of 0, or false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
lang (Language ID)	Specifies the language to be used when the generating application is displaying the user interface controls. If this attribute is omitted, than the generating application can select a language of its choice. The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).
noProof (No Proofing)	Specifies that a run of text has been selected by the user to not be checked for mistakes. Therefore if there are spelling, grammar, etc mistakes within this text the generating application should ignore them. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
normalizeH (Normalize Heights)	Specifies the normalization of height that is to be applied to the text run. This is a render-only modification and does not affect the actual characters stored in the text run. This attribute is also distinct from the toggle function where the actual characters stored in the text run are changed. If this attribute is omitted, than a value of 0, or false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
smtClean (SmartTag Clean)	Specifies whether or not a text run has been checked for smart tags. This attribute acts much like the dirty attribute dose for the checking of spelling, grammar, etc. A value of true here indicates to the generating application that this text run should be checked for smart tags. If this attribute is omitted, than a value of 0, or false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

Attributes	Description

smtId (SmartTag ID)	Specifies a smart tag identifier for a run of text. This ID is unique throughout the presentation and is used to reference corresponding auxiliary information about the smart tag. [*Note*: For a complete definition of smart tags, which are semantically identical throughout Office Open XML, see §17.5.1. *end note*]
	
	[*Example*: Consider the following DrawingML:
	
	`<p:txBody>`
	`  <a:bodyPr/>`
	`  <a:lstStyle/>`
	`  <a:p>`
	`  <a:r>`
	`  <a:rPr lang="en-US" dirty="0" smtId="1"/>`
	`  <a:t>CNTS</a:t>`
	`  </a:r>`
	`  <a:endParaRPr lang="en-US" dirty="0"/>`
	`  </a:p>`
	`</p:txBody>`
	
	The text run has a smtId attribute value of 1, which denotes that the text should be inspected for smart tag information, which in this case maps to a stock ticker symbol. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

spc (Spacing)	Specifies the spacing between characters within a text run. This spacing is specified numerically and should be consistently applied across the entire run of text by the generating application. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. If this attribute is omitted than a value of 0 or no adjustment is assumed. The possible values for this attribute are defined by the ST_TextPoint simple type (§20.1.10.73).

strike (Strikethrough)	Specifies whether a run of text is formatted as strikethrough text. If this attribute is omitted, than no strikethrough is assumed.
	
	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`  …`
	`  <a:rPr strike="sngStrike"/>`
	`  …`
	`  <a:t>Some Text</a:t>`
	`  …`
	`</a:p>`

Attributes	Description
	The above run of text is formatted as single strikethrough text. *end example*] The possible values for this attribute are defined by the ST_TextStrikeType simple type (§20.1.10.78).

sz (Font Size)	Specifies the size of text within a text run. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. If this attribute is omitted, than the value in defRPr should be used.
	
	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`…`
	`<a:rPr sz="1200"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	
	The above run of text is formatted with a 12 point text size. *end example*]
	
	The possible values for this attribute are defined by the ST_TextFontSize simple type (§20.1.10.67).

u (Underline)	Specifies whether a run of text is formatted as underlined text. If this attribute is omitted, than no underline is assumed.
	
	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`…`
	`<a:rPr u="sng"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	
	The above run of text is formatted as single underline text. *end example*]
	
	The possible values for this attribute are defined by the ST_TextUnderlineType simple type (§20.1.10.81).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextCharacterProperties) is located in §A.4.1. *end note*]

21.1.2.3.3 ea (East Asian Font)

This element specifies that an East Asian font be used for a specific run of text. This font is specified with a typeface attribute much like the others but is specifically classified as an East Asian font.

[*Example*: Consider the DrawingML shown below.

```
<a:r>
  <a:rPr …>
    <a:ea typeface="Sample Font"/>
  </a:rPr>
  <a:t>Sample Text</a:t>
</a:r>
```

The above run of text is rendered using the East Asian font "Sample Font". *end example*]

If the specified font is not available on a system being used for rendering, then the attributes of this element can be utilized to select an alternative font.

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>charset (Similar Character Set)</td><td>Specifies the character set which 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:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x00</td><td>Specifies the ANSI character set. (IANA name iso-8859-1)</td></tr>
<tr><td>0x01</td><td>Specifies the default character set.</td></tr>
<tr><td>0x02</td><td>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.</td></tr>
<tr><td>0x4D</td><td>Specifies a Macintosh (Standard Roman) character set. (IANA name macintosh)</td></tr>
<tr><td>0x80</td><td>Specifies the JIS character set. (IANA name shift_jis)</td></tr>
<tr><td>0x81</td><td>Specifies the Hangul character set. (IANA name ks_c_5601-1987)</td></tr>
<tr><td>0x82</td><td>Specifies a Johab character set. (IANA name KS C-5601-1992)</td></tr>
<tr><td>0x86</td><td>Specifies the GB-2312 character set. (IANA name GBK)</td></tr>
<tr><td>0x88</td><td>Specifies the Chinese Big Five character set. (IANA name Big5)</td></tr>
<tr><td>0xA1</td><td>Specifies a Greek character set. (IANA name windows-1253)</td></tr>
<tr><td>0xA2</td><td>Specifies a Turkish character set. (IANA name iso-8859-9)</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>0xA3</td><td>Specifies a Vietnamese character set. (IANA name windows-1258)</td></tr>
<tr><td>0xB1</td><td>Specifies a Hebrew character set. (IANA name windows-1255)</td></tr>
<tr><td>0xB2</td><td>Specifies an Arabic character set. (IANA name windows-1256)</td></tr>
<tr><td>0xBA</td><td>Specifies a Baltic character set. (IANA name windows-1257)</td></tr>
<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)</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 (§22.9.2.8).</td></tr>
<tr><td>pitchFamily (Similar Font Family)</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>
</table>
</td></tr>
</table>

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<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>
<tr><td>0x52</td><td>VARIABLE PITCH + DECORATIVE FONT FAMILY</td></tr>
</table>

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.</td></tr>
<tr><td>typeface (Text Typeface)</td><td>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 (§20.1.10.80).</td></tr>
</table>

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextFont) is located in §A.4.1. *end note*]

21.1.2.3.4 highlight (Highlight Color)

This element specifies the highlight color that is present for a run of text.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:r>
      <a:rPr …>
        <a:highlight>
          <a:srgbClr val="FFFF00"/>
        </a:highlight>
      </a:rPr>
```



```
        …
        <a:t>Sample Text</a:t>
        …
      </a:r>
    </a:p>
    …
  </p:txBody>
```

The above run of text has a yellow highlight color as specified by the srgbClr child element. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

21.1.2.3.5 hlinkClick (Click Hyperlink)

Specifies the on-click hyperlink information to be applied to a run of text. When the hyperlink text is clicked the link is fetched.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:r>
      <a:rPr …>
        <a:hlinkClick r:id="rId2" tooltip="Some Sample Text"/>
      </a:rPr>
      …
      <a:t>Sample Text</a:t>
      …
    </a:r>
  </a:p>
  …
</p:txBody>
```

The above run of text is a hyperlink that points to the resource pointed at by rId2 within this slides relationship file. Additionally this text should display a tooltip when the mouse is hovered over the run of text. *end example*]

Attributes	Description
action (Action Setting)	Specifies an action that is to be taken when this hyperlink is activated. This can be used to specify a slide to be navigated to or a script of code to be run. The possible values for this attribute are defined by the W3C XML Schema string datatype.
endSnd (End Sounds)	Specifies if the URL in question should stop all sounds that are playing when it is clicked.

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
highlightClick (Highlight Click)	Specifies if this attribute has already been used within this document. That is when a hyperlink has already been visited that this attribute would be utilized so the generating application can determine the color of this text. If this attribute is omitted, then a value of 0 or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
history (Add Hyperlink to Page History)	Specifies whether to add this URI to the history when navigating to it. This allows for the viewing of this presentation without the storing of history information on the viewing machine. If this attribute is omitted, then a value of `1` or `true` is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Drawing Object Hyperlink Target) Namespace: http://purl.oclc.org/ooxml/officeDocument/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 (§22.8.2.1).
invalidUrl (Invalid URL)	Specifies the URL when it has been determined by the generating application that the URL is invalid. That is the generating application can still store the URL but it is known that this URL is not correct. The possible values for this attribute are defined by the W3C XML Schema string datatype.
tgtFrame (Target Frame)	Specifies the target frame that is to be used when opening this hyperlink. When the hyperlink is activated this attribute is used to determine if a new window is launched for viewing or if an existing one can be used. If this attribute is omitted, than a new window is opened. The possible values for this attribute are defined by the W3C XML Schema string datatype.
tooltip (Hyperlink Tooltip)	Specifies the tooltip that should be displayed when the hyperlink text is hovered over with the mouse. If this attribute is omitted, than the hyperlink text itself can be displayed. 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_Hyperlink) is located in §A.4.1. *end note*]

21.1.2.3.6 hlinkMouseOver (Mouse-Over Hyperlink)

Specifies the mouse-over hyperlink information to be applied to a run of text. When the mouse is hovered over this hyperlink text the link is fetched.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:r>
      <a:rPr …>
        <a:hlinkMouseOver r:id="rId2" tooltip="Some Sample Text"/>
      </a:rPr>
      …
      <a:t>Sample Text</a:t>
      …
    </a:r>
  </a:p>
  …
</p:txBody>
```

The above run of text is a hyperlink that points to the resource pointed at by rId2 within this slides relationship file. Additionally this text should display a tooltip when the mouse is hovered over the run of text. *end example*]

Attributes	Description
action (Action Setting)	Specifies an action that is to be taken when this hyperlink is activated. This can be used to specify a slide to be navigated to or a script of code to be run. The possible values for this attribute are defined by the W3C XML Schema string datatype.
endSnd (End Sounds)	Specifies if the URL in question should stop all sounds that are playing when it is clicked. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
highlightClick (Highlight Click)	Specifies if this attribute has already been used within this document. That is when a hyperlink has already been visited that this attribute would be utilized so the generating application can determine the color of this text. If this attribute is omitted, then a value of 0 or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

Attributes	Description
history (Add Hyperlink to Page History)	Specifies whether to add this URI to the history when navigating to it. This allows for the viewing of this presentation without the storing of history information on the viewing machine. If this attribute is omitted, then a value of `1` or `true` is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Drawing Object Hyperlink Target) Namespace: http://purl.oclc.org/ooxml/officeDocument/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 (§22.8.2.1).
invalidUrl (Invalid URL)	Specifies the URL when it has been determined by the generating application that the URL is invalid. That is the generating application can still store the URL but it is known that this URL is not correct. The possible values for this attribute are defined by the W3C XML Schema string datatype.
tgtFrame (Target Frame)	Specifies the target frame that is to be used when opening this hyperlink. When the hyperlink is activated this attribute is used to determine if a new window is launched for viewing or if an existing one can be used. If this attribute is omitted, than a new window is opened. The possible values for this attribute are defined by the W3C XML Schema string datatype.
tooltip (Hyperlink Tooltip)	Specifies the tooltip that should be displayed when the hyperlink text is hovered over with the mouse. If this attribute is omitted, than the hyperlink text itself can be displayed. 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_Hyperlink) is located in §A.4.1. *end note*]

21.1.2.3.7 latin (Latin Font)

This element specifies that a Latin font be used for a specific run of text. This font is specified with a typeface attribute much like the others but is specifically classified as a Latin font.

[*Example*: Consider the DrawingML shown below.


```
<a:r>
  <a:rPr …>
    <a:latin typeface="Sample Font"/>
  </a:rPr>
  <a:t>Sample Text</a:t>
</a:r>
```

The above run of text is rendered using the Latin font "Sample Font". *end example*]

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>charset (Similar Character Set)</td><td>Specifies the character set which 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:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x00</td><td>Specifies the ANSI character set. (IANA name iso-8859-1)</td></tr>
<tr><td>0x01</td><td>Specifies the default character set.</td></tr>
<tr><td>0x02</td><td>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.</td></tr>
<tr><td>0x4D</td><td>Specifies a Macintosh (Standard Roman) character set. (IANA name macintosh)</td></tr>
<tr><td>0x80</td><td>Specifies the JIS character set. (IANA name shift_jis)</td></tr>
<tr><td>0x81</td><td>Specifies the Hangul character set. (IANA name ks_c_5601-1987)</td></tr>
<tr><td>0x82</td><td>Specifies a Johab character set. (IANA name KS C-5601-1992)</td></tr>
<tr><td>0x86</td><td>Specifies the GB-2312 character set. (IANA name GBK)</td></tr>
<tr><td>0x88</td><td>Specifies the Chinese Big Five character set. (IANA name Big5)</td></tr>
<tr><td>0xA1</td><td>Specifies a Greek character set. (IANA name windows-1253)</td></tr>
<tr><td>0xA2</td><td>Specifies a Turkish character set. (IANA name iso-8859-9)</td></tr>
<tr><td>0xA3</td><td>Specifies a Vietnamese character set. (IANA name windows-1258)</td></tr>
<tr><td>0xB1</td><td>Specifies a Hebrew character set. (IANA name windows-1255)</td></tr>
<tr><td>0xB2</td><td>Specifies an Arabic character set. (IANA name windows-1256)</td></tr>
<tr><td>0xBA</td><td>Specifies a Baltic character set. (IANA name windows-1257)</td></tr>
<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-</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>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)</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 (§22.9.2.8).</td></tr>
<tr><td>pitchFamily (Similar Font Family)</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 an octet value 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>
<tr><td>0x52</td><td>VARIABLE PITCH + DECORATIVE FONT FAMILY</td></tr>
</table>

This information is determined by querying the font when present and shall not be</td></tr>
</table>

Attributes	Description
	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. [*Note*: Although the attribute name is pitchFamily, the integer value of this attribute specifies the font family with higher 4 bits and the font pitch with lower 4 bits. *end note*] The possible values for this attribute are defined by the W3C XML Schema byte datatype.
typeface (Text Typeface)	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 (§20.1.10.80).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextFont) is located in §A.4.1. *end note*]

21.1.2.3.8 r (Text Run)

This element specifies the presence of a run of text within the containing text body. The run element is the lowest level text separation mechanism within a text body. A text run can contain text run properties associated with the run. If no properties are listed then properties specified in the defRPr element are used.

[*Example*: Consider the case where the user would like to describe a text body that contains two runs of text and would like one to be bold and the other not. The following DrawingML would specify such a text body.

```
<p:txBody>
  …
  <a:r>
    <a:rPr b="1">
    </a:rPr>
    <a:t>Some text</a:t>
  </a:r>
  …
  <a:r>
    <a:rPr/>
    <a:t>Some text</a:t>
  </a:r>
</p:txBody>
```

The above text body has the first run be formatted bold and the second normally. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_RegularTextRun) is located in §A.4.1. *end note*]

21.1.2.3.9 rPr (Text Run Properties)

This element contains all run level text properties for the text runs within a containing paragraph.

[*Example*: Consider the DrawingML shown below.

```
<a:p>
  …
  <a:rPr u="sng"/>
  …
  <a:t>Some Text</a:t>
  …
</a:p>
```

The run of text described above is formatting with a single underline of text matching color. *end example*]

Attributes	Description
altLang (Alternative Language)	Specifies the alternate language to use when the generating application is displaying the user interface controls. If this attribute is omitted, than the lang attribute is used here. The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).
b (Bold)	Specifies whether a run of text is formatted as bold text. If this attribute is omitted, than a value of 0, or false is assumed. [*Example*: Consider the DrawingML shown below. <a:p> … <a:rPr b="1"/> … <a:t>Some Text</a:t> … </a:p> The above run of text is formatted as bold text. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
baseline (Baseline)	Specifies the baseline for both the superscript and subscript fonts. The size is specified using a percentage where 1% is equal to 1 percent of the font size and 100% is equal to 100 percent font of the font size. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
bmk (Bookmark Link Target)	Specifies the link target name that is used to reference to the proper link properties in a custom XML part within the document.

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.
cap (Capitalization)	Specifies the capitalization that is to be applied to the text run. This is a render-only modification and does not affect the actual characters stored in the text run. This attribute is also distinct from the toggle function where the actual characters stored in the text run are changed. The possible values for this attribute are defined by the ST_TextCapsType simple type (§20.1.10.63).
dirty (Dirty)	Specifies that the content of a text run has changed since the proofing tools have last been run. Effectively this flags text that is to be checked again by the generating application for mistakes such as spelling, grammar, etc. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
err (Spelling Error)	Specifies that when this run of text was checked for spelling, grammar, etc. that a mistake was indeed found. This allows the generating application to effectively save the state of the mistakes within the document instead of having to perform a full pass check upon opening the document. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

i (Italics)	Specifies whether a run of text is formatted as italic text. If this attribute is omitted, than a value of 0, or false is assumed.
	
	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`…`
	`<a:rPr i="1"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	
	The above run of text is formatted as italic text. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

kern (Kerning)	Specifies the minimum font size at which character kerning occurs for this text run. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. If this attribute is omitted, than kerning occurs for all font sizes down to a 0 point font.

Attributes	Description
	The possible values for this attribute are defined by the ST_TextNonNegativePoint simple type (§20.1.10.72).
kumimoji (Kumimoji)	Specifies whether the numbers contained within vertical text continue vertically with the text or whether they are to be displayed horizontally while the surrounding characters continue in a vertical fashion. If this attribute is omitted, than a value of 0, or false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
lang (Language ID)	Specifies the language to be used when the generating application is displaying the user interface controls. If this attribute is omitted, than the generating application can select a language of its choice. The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).
noProof (No Proofing)	Specifies that a run of text has been selected by the user to not be checked for mistakes. Therefore if there are spelling, grammar, etc mistakes within this text the generating application should ignore them. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
normalizeH (Normalize Heights)	Specifies the normalization of height that is to be applied to the text run. This is a render-only modification and does not affect the actual characters stored in the text run. This attribute is also distinct from the toggle function where the actual characters stored in the text run are changed. If this attribute is omitted, than a value of 0, or false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
smtClean (SmartTag Clean)	Specifies whether or not a text run has been checked for smart tags. This attribute acts much like the dirty attribute dose for the checking of spelling, grammar, etc. A value of true here indicates to the generating application that this text run should be checked for smart tags. If this attribute is omitted, than a value of 0, or false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

smtId (SmartTag ID)	Specifies a smart tag identifier for a run of text. This ID is unique throughout the presentation and is used to reference corresponding auxiliary information about the smart tag. [*Note*: For a complete definition of smart tags, which are semantically identical throughout Office Open XML, see §17.5.1. *end note*]
	
	[*Example*: Consider the following DrawingML:
	
	`<p:txBody>`
	`  <a:bodyPr/>`

Attributes	Description
	<a:lstStyle/> <a:p> <a:r> <a:rPr lang="en-US" dirty="0" smtId="1"/> <a:t>CNTS</a:t> </a:r> <a:endParaRPr lang="en-US" dirty="0"/> </a:p> </p:txBody> The text run has a smtId attribute value of 1, which denotes that the text should be inspected for smart tag information, which in this case maps to a stock ticker symbol. *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
spc (Spacing)	Specifies the spacing between characters within a text run. This spacing is specified numerically and should be consistently applied across the entire run of text by the generating application. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. If this attribute is omitted than a value of 0 or no adjustment is assumed. The possible values for this attribute are defined by the ST_TextPoint simple type (§20.1.10.73).
strike (Strikethrough)	Specifies whether a run of text is formatted as strikethrough text. If this attribute is omitted, than no strikethrough is assumed. [*Example*: Consider the DrawingML shown below. <a:p> … <a:rPr strike="sngStrike"/> … <a:t>Some Text</a:t> … </a:p> The above run of text is formatted as single strikethrough text. *end example*] The possible values for this attribute are defined by the ST_TextStrikeType simple type (§20.1.10.78).
sz (Font Size)	Specifies the size of text within a text run. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. If this attribute is omitted, than the value in defRPr should be used.

Attributes	Description

	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`…`
	`<a:rPr sz="1200"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	
	The above run of text is formatted with a 12 point text size. *end example*]
	
	The possible values for this attribute are defined by the ST_TextFontSize simple type (§20.1.10.67).

u (Underline)	Specifies whether a run of text is formatted as underlined text. If this attribute is omitted, than no underline is assumed.
	
	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`…`
	`<a:rPr u="sng"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	
	The above run of text is formatted as single underline text. *end example*]
	
	The possible values for this attribute are defined by the ST_TextUnderlineType simple type (§20.1.10.81).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextCharacterProperties) is located in §A.4.1. *end note*]

21.1.2.3.10 sym (Symbol Font)

This element specifies that a symbol font be used for a specific run of text. This font is specified with a typeface attribute much like the others but is specifically classified as a symbol font.

[*Example*: Consider the DrawingML shown below.


```
<a:r>
  <a:rPr …>
    <a:sym typeface="Sample Font"/>
  </a:rPr>
  <a:t>Sample Text</a:t>
</a:r>
```

The above run of text is rendered using the symbol font "Sample Font". *end example*]

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>charset (Similar Character Set)</td><td>Specifies the character set which 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:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x00</td><td>Specifies the ANSI character set. (IANA name iso-8859-1)</td></tr>
<tr><td>0x01</td><td>Specifies the default character set.</td></tr>
<tr><td>0x02</td><td>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.</td></tr>
<tr><td>0x4D</td><td>Specifies a Macintosh (Standard Roman) character set. (IANA name macintosh)</td></tr>
<tr><td>0x80</td><td>Specifies the JIS character set. (IANA name shift_jis)</td></tr>
<tr><td>0x81</td><td>Specifies the Hangul character set. (IANA name ks_c_5601-1987)</td></tr>
<tr><td>0x82</td><td>Specifies a Johab character set. (IANA name KS C-5601-1992)</td></tr>
<tr><td>0x86</td><td>Specifies the GB-2312 character set. (IANA name GBK)</td></tr>
<tr><td>0x88</td><td>Specifies the Chinese Big Five character set. (IANA name Big5)</td></tr>
<tr><td>0xA1</td><td>Specifies a Greek character set. (IANA name windows-1253)</td></tr>
<tr><td>0xA2</td><td>Specifies a Turkish character set. (IANA name iso-8859-9)</td></tr>
<tr><td>0xA3</td><td>Specifies a Vietnamese character set. (IANA name windows-1258)</td></tr>
<tr><td>0xB1</td><td>Specifies a Hebrew character set. (IANA name windows-1255)</td></tr>
<tr><td>0xB2</td><td>Specifies an Arabic character set. (IANA name windows-1256)</td></tr>
<tr><td>0xBA</td><td>Specifies a Baltic character set. (IANA name windows-1257)</td></tr>
<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-</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>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)</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 (§22.9.2.8).</td></tr>
<tr><td>pitchFamily (Similar Font Family)</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 an octet value 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>
<tr><td>0x52</td><td>VARIABLE PITCH + DECORATIVE FONT FAMILY</td></tr>
</table>

This information is determined by querying the font when present and shall not be</td></tr>
</table>

Attributes	Description
	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. [*Note*: Although the attribute name is pitchFamily, the integer value of this attribute specifies the font family with higher 4 bits and the font pitch with lower 4 bits. *end note*] The possible values for this attribute are defined by the W3C XML Schema byte datatype.
typeface (Text Typeface)	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 (§20.1.10.80).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextFont) is located in §A.4.1. *end note*]

21.1.2.3.11 t (Text String)

This element specifies the actual text for this text run. This is the text that is formatted using all specified body, paragraph and run properties. This element shall be present within a run of text.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    …
    <a:r>
      …
      <a:t>Sample Text</a:t>
      …
    </a:r>
    …
  </a:p>
  …
</p:txBody>
```

The above DrawingML specifies a text body containing a single paragraph, containing a single run which contains the actual text specified with the <a:t> element.

The possible values for this element are defined by the W3C XML Schema string datatype.

21.1.2.3.12 uFill (Underline Fill)

This element specifies the fill color of an underline for a run of text.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:r>
      <a:rPr …>
        <a:uFill>
          <a:solidFill>
            <a:srgbClr val="FFFF00"/>
          </a:solidFill>
        </a:uFill>
      </a:rPr>
      …
      <a:t>Sample Text</a:t>
      …
    </a:r>
  </a:p>
  …
</p:txBody>
```

The underline color of the above text is yellow specified by the srgbClr child element. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextUnderlineFillGroupWrapper) is located in §A.4.1. *end note*]

21.1.2.3.13 uFillTx (Underline Fill Properties Follow Text)

This element specifies that the fill color of an underline for a run of text should be of the same color as the text run within which it is contained.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:r>
      <a:rPr …>
        <a:uFillTx>
      </a:rPr>
      …
```



```
        <a:t>Sample Text</a:t>
        …
      </a:r>
    </a:p>
    …
  </p:txBody>
```

The underline color of the above text follows the color of the text run within which it resides. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextUnderlineFillFollowText) is located in §A.4.1. *end note*]

21.1.2.3.14 uLn (Underline Stroke)

This element specifies the properties for the stroke of the underline that is present within a run of text.

[*Example*: Consider the DrawingML shown below.

```
  <p:txBody>
    …
    <a:p>
      <a:r>
        <a:rPr …>
          <a:uLn algn="r">
        </a:rPr>
        …
        <a:t>Sample Text</a:t>
        …
      </a:r>
    </a:p>
    …
  </p:txBody>
```

The underline alignment of the above text is right aligned. *end example*]

Attributes	Description
algn (Stroke Alignment)	Specifies the alignment to be used for the underline stroke. The possible values for this attribute are defined by the ST_PenAlignment simple type (§20.1.10.39).
cap (Line Ending Cap Type)	Specifies the ending caps that should be used for this line. [*Note*: Examples of cap types are rounded, flat, etc. *end note*] If this attribute is omitted, than a value of square is assumed. The possible values for this attribute are defined by the ST_LineCap simple type (§20.1.10.31).

Attributes	Description
cmpd (Compound Line Type)	Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. The possible values for this attribute are defined by the ST_CompoundLine simple type (§20.1.10.15).
w (Line Width)	Specifies the width to be used for the underline stroke. If this attribute is omitted, then a value of 0 is assumed. The possible values for this attribute are defined by the ST_LineWidth simple type (§20.1.10.35).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineProperties) is located in §A.4.1. *end note*]

21.1.2.3.15 uLnTx (Underline Follows Text)

This element specifies that the stroke style of an underline for a run of text should be of the same as the text run within which it is contained.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:r>
      <a:rPr …>
        <a:uLnTx>
      </a:rPr>
      …
      <a:t>Sample Text</a:t>
      …
    </a:r>
  </a:p>
  …
</p:txBody>
```

The underline stroke of the above text follows the stroke of the run text within which it resides. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextUnderlineLineFollowText) is located in §A.4.1. *end note*]

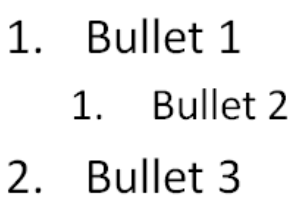
21.1.2.4 Bullets and Numbering

In addition to the above body, paragraph and text run properties there can also be a structure of bullets and numbering that can be defined by utilizing a few of these layers. Since Bullet and Numbering does span multiple formatting levels it is described on it's own in the following section.

21.1.2.4.1 buAutoNum (Auto-Numbered Bullet)

This element specifies that automatic numbered bullet points should be applied to a paragraph. These are not just numbers used as bullet points but instead automatically assigned numbers that are based on both buAutoNum attributes and paragraph level.

[*Example*: Consider the DrawingML content shown below.




```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buAutoNum type="arabicPeriod"/>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  <a:p>
    <a:pPr lvl="1"…>
      <a:buAutoNum type="arabicPeriod"/>
    </a:pPr>
    …
    <a:t>Bullet 2</a:t>
    …
  </a:p>
```

```
    <a:p>
      <a:pPr …>
        <a:buAutoNum type="arabicPeriod"/>
      </a:pPr>
      …
      <a:t>Bullet 3</a:t>
      …
    </a:p>
    …
  </p:txBody>
```

For the above text there are a total of three bullet points. Two of which are at lvl="0" and one at lvl="1". Due to this breakdown of levels, the numbering sequence that should be automatically applied is 1, 1, 2 as is shown in the picture above. *end example*]

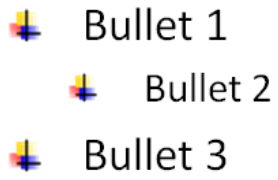
Attributes	Description
startAt (Start Numbering At)	Specifies the number that starts a given sequence of automatically numbered bullets. When the numbering is alphabetical, the number should map to the appropriate letter. For instance 1 maps to 'a', 2 to 'b' and so on. If the numbers are larger than 26, then multiple letters should be used. For instance 27 should be represented as 'aa' and similarly 53 should be 'aaa'. The possible values for this attribute are defined by the ST_TextBulletStartAtNum simple type (§20.1.10.62).
type (Bullet Autonumbering Type)	Specifies the numbering scheme that is to be used. This allows for the describing of formats other than strictly numbers. For instance, a set of bullets can be represented by a series of Roman numerals instead of the standard 1,2,3,etc. number set. The possible values for this attribute are defined by the ST_TextAutonumberScheme simple type (§20.1.10.60).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextAutonumberBullet) is located in §A.4.1. *end note*]

21.1.2.4.2 buBlip (Picture Bullet)

This element specifies that a picture be applied to a set of bullets. This element allows for any standard picture format graphic to be used instead of the typical bullet characters. This opens up the possibility for bullets to be anything the generating application would seek to apply.

[*Example*: Consider the DrawingML shown below.




```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buBlip>
        <a:blip r:embed="rId2"/>
      </a:buBlip>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  <a:p>
    <a:pPr lvl="1"…>
      <a:buBlip>
        <a:blip r:embed="rId2"/>
      </a:buBlip>
    </a:pPr>
    …
    <a:t>Bullet 2</a:t>
    …
  </a:p>
  <a:p>
    <a:pPr …>
      <a:buBlip>
        <a:blip r:embed="rId2"/>
      </a:buBlip>
    </a:pPr>
    …
    <a:t>Bullet 3</a:t>
    …
  </a:p>
  …
</p:txBody>
```

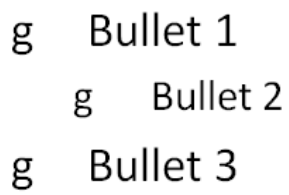
For the above text there are a total of three bullet points. Two of which are at lvl="0" and one at lvl="1". Because the same picture is specified for each bullet the levels do not stand out here. The only difference is the indentation as shown in the picture above. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBlipBullet) is located in §A.4.1. *end note*]

21.1.2.4.3 buChar (Character Bullet)

This element specifies that a character be applied to a set of bullets. These bullets are allowed to be any character in any font that the system is able to support. If no bullet font is specified along with this element then the paragraph font is used.

[*Example*: Consider the DrawingML shown below.




```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buFont typeface="Calibri"/>
      <a:buChar char="g"/>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  <a:p>
    <a:pPr lvl="1"…>
      <a:buFont typeface="Calibri"/>
      <a:buChar char="g"/>
    </a:pPr>
    …
    <a:t>Bullet 2</a:t>
    …
  </a:p>
```



```
    <a:p>
      <a:pPr …>
        <a:buFont typeface="Calibri"/>
        <a:buChar char="g"/>
      </a:pPr>
      …
      <a:t>Bullet 3</a:t>
      …
    </a:p>
    …
  </p:txBody>
```

For the above text there are a total of three bullet points. Two of which are at lvl="0" and one at lvl="1". Because the same character is specified for each bullet the levels do not stand out here. The only difference is the indentation as shown in the picture above. *end example*]

Attributes	Description
char (Bullet Character)	Specifies the character to be used in place of the standard bullet point. This character can be any character for the specified font that is supported by the system upon which this document is being viewed. 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_TextCharBullet) is located in §A.4.1. *end note*]

21.1.2.4.4 buClr (Color Specified)

This element specifies the color to be used on bullet characters within a given paragraph. The color is specified using the numerical RGB color format.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buClr>
        <a:srgbClr val="FFFF00"/>
      </a:buClr>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The color of the above bullet does not follow the text color but instead has a yellow color specified by val="FFFF00". This color should only apply to the actual bullet character and not to the text within the bullet. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

21.1.2.4.5 buClrTx (Follow Text)

This element specifies that the color of the bullets for a paragraph should be of the same color as the text run within which each bullet is contained.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buClrTx>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The color of the above bullet follows the default text color of the text for the run of text shown above since no specific text color was specified. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBulletColorFollowText) is located in §A.4.1. *end note*]

21.1.2.4.6 buFont (Specified)

This element specifies the font to be used on bullet characters within a given paragraph. The font is specified using the typeface that it is registered as within the generating application.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buFont typeface="Arial"/>
      <a:buChar char="g"/>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The font of the above bullet does not follow the text font but instead has Arial font specified by typeface="Arial". This font should only apply to the actual bullet character and not to the text within the bullet. *end example*]

Attributes	Description
charset (Similar Character Set)	Specifies the character set which 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

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>macintosh)</td></tr>
<tr><td>0x80</td><td>Specifies the JIS character set. (IANA name shift_jis)</td></tr>
<tr><td>0x81</td><td>Specifies the Hangul character set. (IANA name ks_c_5601-1987)</td></tr>
<tr><td>0x82</td><td>Specifies a Johab character set. (IANA name KS C-5601-1992)</td></tr>
<tr><td>0x86</td><td>Specifies the GB-2312 character set. (IANA name GBK)</td></tr>
<tr><td>0x88</td><td>Specifies the Chinese Big Five character set. (IANA name Big5)</td></tr>
<tr><td>0xA1</td><td>Specifies a Greek character set. (IANA name windows-1253)</td></tr>
<tr><td>0xA2</td><td>Specifies a Turkish character set. (IANA name iso-8859-9)</td></tr>
<tr><td>0xA3</td><td>Specifies a Vietnamese character set. (IANA name windows-1258)</td></tr>
<tr><td>0xB1</td><td>Specifies a Hebrew character set. (IANA name windows-1255)</td></tr>
<tr><td>0xB2</td><td>Specifies an Arabic character set. (IANA name windows-1256)</td></tr>
<tr><td>0xBA</td><td>Specifies a Baltic character set. (IANA name windows-1257)</td></tr>
<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)</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 (§22.9.2.8).</td></tr>
<tr><td>pitchFamily (Similar Font Family)</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 an octet value 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>
</table>
</td></tr>
</table>

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<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>
<tr><td>0x52</td><td>VARIABLE PITCH + DECORATIVE FONT FAMILY</td></tr>
</table>
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.
[Note: Although the attribute name is pitchFamily, the integer value of this attribute specifies the font family with higher 4 bits and the font pitch with lower 4 bits. end note]
The possible values for this attribute are defined by the W3C XML Schema byte datatype.
</td></tr>
<tr><td>typeface (Text Typeface)</td><td>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 (§20.1.10.80).</td></tr>
</table>

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextFont) is located in §A.4.1. *end note*]

21.1.2.4.7 buFontTx (Follow text)

This element specifies that the font of the bullets for a paragraph should be of the same font as the text run within which each bullet is contained.

[*Example*: Consider the DrawingML shown below.


```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buFontTx>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The font of the above bullet follows the default text font of the text for the run of text shown above since no specific text font was specified. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBulletTypefaceFollowText) is located in §A.4.1. *end note*]

21.1.2.4.8 buNone (No Bullet)

This element specifies that the paragraph within which it is applied is to have no bullet formatting applied to it. That is to say that there should be no bulleting found within the paragraph where this element is specified.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buNone/>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The above paragraph is formatted with no bullets. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextNoBullet) is located in §A.4.1. *end note*]

21.1.2.4.9 buSzPct (Bullet Size Percentage)

This element specifies the size in percentage of the surrounding text to be used on bullet characters within a given paragraph.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buSzPct val="111%"/>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The size of the above bullet follows the text size in that it is always rendered at 111% the size of the text within the given text run. This is specified by val="111%", with a restriction on the values not being less than 25% or more than 400%. This percentage size should only apply to the actual bullet character and not to the text within the bullet. *end example*]

Attributes	Description
val (Value)	Specifies the percentage of the text size that this bullet should be. This attribute should not be lower than 25% and not be higher than 400%. The possible values for this attribute are defined by the ST_TextBulletSizePercent simple type (§20.1.10.61).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBulletSizePercent) is located in §A.4.1. *end note*]

21.1.2.4.10 buSzPts (Bullet Size Points)

This element specifies the size in points to be used on bullet characters within a given paragraph. The size is specified using the points where 100 is equal to 1 point font and 1200 is equal to 12 point font.

[*Example*: Consider the DrawingML shown below.


```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buSzPts val="1400"/>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The size of the above bullet does not follow the text size of the text within the given text run. The bullets size is specified by val="1400", which corresponds to a point size of 14. This bullet size should only apply to the actual bullet character and not to the text within the bullet. *end example*]

Attributes	Description
val (Value)	Specifies the size of the bullets in point size. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. The possible values for this attribute are defined by the ST_TextFontSize simple type (§20.1.10.67).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBulletSizePoint) is located in §A.4.1. *end note*]

21.1.2.4.11 buSzTx (Bullet Size Follows Text)

This element specifies that the size of the bullets for a paragraph should be of the same point size as the text run within which each bullet is contained.

[*Example*: Consider the DrawingML shown below.


```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buSzTx>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The size of the above bullet follows the default text size of the text for the run of text shown above since no specific text size was specified. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBulletSizeFollowText) is located in §A.4.1. *end note*]

21.1.2.4.12 lstStyle (Text List Styles)

This element specifies the list of styles associated with this body of text.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextListStyle) is located in §A.4.1. *end note*]

21.1.2.4.13 lvl1pPr (List Level 1 Text Style)

This element specifies all paragraph level text properties for all elements that have the attribute lvl="0". There are a total of 9 level text property elements allowed, levels 0-8. It is recommended that the order in which this and other level property elements are specified be in order of increasing level. That is lvl2pPr should come before lvl3pPr. This allows the lower level properties to take precedence over the higher level ones because they are parsed first.

[*Example*: Consider the following DrawingML code that would specify a paragraph to follow the level style defined in lvl1pPr and thus create a paragraph of text that has no bullets and is right aligned.

```
<p:txBody>
  …
  <a:lstStyle>
    <a:lvl1pPr algn="r">
      <a:buNone/>
    </a:lvl1pPr>
  </a:lstStyle>
  <a:p>
    <a:pPr lvl="0">
    </a:pPr>
    …
    <a:t>Some text</a:t>
    …
  </a:p>
</p:txBody>
```

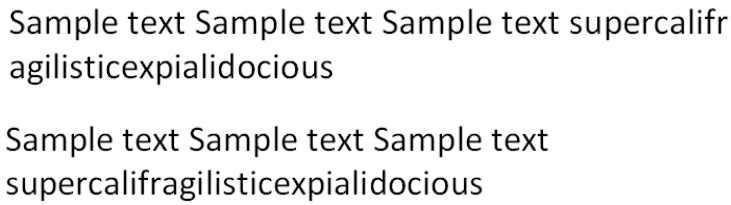
end example]

[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]

Attributes	Description
algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied. Sample text [*Example*: Consider the case where the user wishes to have two columns of text that have a justified alignment, much like text within a book. The following DrawingML could describe this.

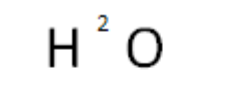
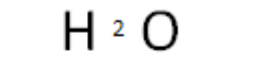
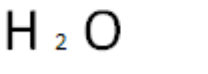
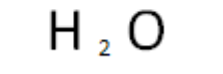
Attributes	Description
	<p:txBody> <a:bodyPr numCol="2" spcCol="914400"…/> <a:normAutofit/> </a:bodyPr> … <a:p> <a:pPr marL="0" algn="just"> <a:buNone/> </a:pPr> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextAlignType simple type (§20.1.10.58).
defTabSz (Default Tab Size)	Specifies the default size for a tab character within this paragraph. This attribute should be used to describe the spacing of tabs within the paragraph instead of a leading indentation tab. For indentation tabs there are the marL and indent attributes to assist with this. [*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this. <p:txBody> … <a:p> <a:pPr defTabSz="376300" …/> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces

Attributes	Description

	without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied.
	
	[*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr eaLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied. [*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below.

Attributes	Description

	
	fontAlgn="t"
	 
	fontAlgn="ctr"
	 
	fontAlgn="base"
	 
	fontAlgn="b"
	
	`<a:txtBody>`
	`…`
	`<a:pPr fontAlgn="b" …/>`
	`…`
	`<a:r>`
	`<a:rPr …/>`
	`<a:t>H </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr sz="1200" …/>`
	`<a:t>2</a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr …/>`
	`<a:t>O</a:t>`
	`</a:r>`
	`…`
	`</p:txBody>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.65).

hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this

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

attribute is omitted, then a value of -342900 is implied.

Here is some Sample text

[*Example*: Consider the scenario where the user now wanted to add a paragraph indentation to the first line of text in their two column format book.

```
<p:txBody>
  <a:bodyPr numCol="2" spcCol="914400"…/>
    <a:normAutofit/>
    </a:bodyPr>
  …
  <a:p>
    <a:pPr marL="0" indent="571500" algn="just">
      <a:buNone/>
    </a:pPr>
    …
    <a:t>Here is some…</a:t>
    …
  </a:p>
</p:txBody>
```

By adding the indent attribute the user has effectively added a first line indent to this paragraph of text. *end example*]

The possible values for this attribute are defined by the ST_TextIndent simple type (§20.1.10.69).

</td></tr>
<tr><td>latinLnBrk (Latin Line Break)</td><td>

Specifies whether a Latin word can be broken in half and wrapped onto the next line without a hyphen being added. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. It is not present within the existence of normal breakable Latin words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or `true` is implied.

[*Example*: Consider the case where the presentation contains a long word that must not

</td></tr>
</table>

Attributes	Description

	be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr latinLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values.
	
	[*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr lvl="1" …/>`
	`  …`
	`  <a:t>Sample text</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	end example]

Attributes	Description
	[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*] The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.70).
marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).

rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this attribute is omitted, then a value of 0, or left-to-right is implied.
	
	[*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.
	
	Testتجربة ⟵ rtl = 0
	تجربةTest ⟵ rtl = 1
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`

Attributes	Description
	<a:t>تجربة</a:t> </a:r> </a:p> <a:p> <a:pPr rtl="1"/> <a:r> <a:rPr> <a:rtl w:val="0"/> </a:rPr> <a:t>Test </a:t> </a:r> <a:r> <a:t>تجربة </a:t> </a:r> </a:p> </p:txBody> *end example*] 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_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.4.14 lvl2pPr (List Level 2 Text Style)

This element specifies all paragraph level text properties for all elements that have the attribute lvl="1". There are a total of 9 level text property elements allowed, levels 0-8. It is recommended that the order in which this and other level property elements are specified be in order of increasing level. That is lvl2pPr should come before lvl3pPr. This allows the lower level properties to take precedence over the higher level ones because they are parsed first.

[*Example*: Consider the following DrawingML code that would specify a paragraph to follow the level style defined in lvl2pPr and thus create a paragraph of text that has no bullets and is right aligned.

```
<p:txBody>
  …
  <a:lstStyle>
    <a:lvl2pPr algn="r">
      <a:buNone/>
    </a:lvl2pPr>
  </a:lstStyle>
```

```
    <a:p>
      <a:pPr lvl="1">
      </a:pPr>
      …
      <a:t>Some text</a:t>
      …
    </a:p>
  </p:txBody>
```

end example]

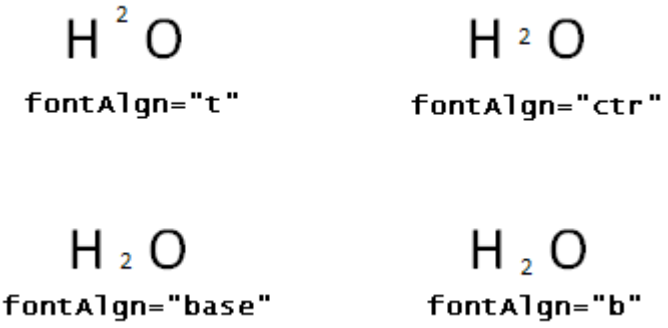
[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]

Attributes	Description
algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied. Sample text [*Example*: Consider the case where the user wishes to have two columns of text that have a justified alignment, much like text within a book. The following DrawingML could describe this. `<p:txBody>` `<a:bodyPr numCol="2" spcCol="914400"…/>` `<a:normAutofit/>` `</a:bodyPr>` … `<a:p>` `<a:pPr marL="0" algn="just">`

Attributes	Description
	<a:buNone/> </a:pPr> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextAlignType simple type (§20.1.10.58).
defTabSz (Default Tab Size)	Specifies the default size for a tab character within this paragraph. This attribute should be used to describe the spacing of tabs within the paragraph instead of a leading indentation tab. For indentation tabs there are the marL and indent attributes to assist with this. [*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this. <p:txBody> … <a:p> <a:pPr defTabSz="376300" …/> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the

Attributes	Description

	long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr eaLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied. [*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below. 

Attributes	Description
	<a:txtBody> … <a:pPr fontAlgn="b" …/> … <a:r> <a:rPr …/> <a:t>H </a:t> </a:r> <a:r> <a:rPr sz="1200" …/> <a:t>2</a:t> </a:r> <a:r> <a:rPr …/> <a:t>O</a:t> </a:r> … </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.65).
hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this attribute is omitted, then a value of -342900 is implied.

Attributes	Description

	Here is some Sample text
	
	[*Example*: Consider the scenario where the user now wanted to add a paragraph indentation to the first line of text in their two column format book.
	
	`<p:txBody>`
	`  <a:bodyPr numCol="2" spcCol="914400"…/>`
	`    <a:normAutofit/>`
	`    </a:bodyPr>`
	`  …`
	`  <a:p>`
	`    <a:pPr marL="0" indent="571500" algn="just">`
	`      <a:buNone/>`
	`    </a:pPr>`
	`    …`
	`    <a:t>Here is some…</a:t>`
	`    …`
	`  </a:p>`
	`</p:txBody>`
	
	By adding the indent attribute the user has effectively added a first line indent to this paragraph of text. *end example*]
	
	The possible values for this attribute are defined by the ST_TextIndent simple type (§20.1.10.69).

latinLnBrk (Latin Line Break)	Specifies whether a Latin word can be broken in half and wrapped onto the next line without a hyphen being added. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. It is not present within the existence of normal breakable Latin words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken

Attributes	Description

	for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr latinLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values.
	
	[*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr lvl="1" …/>`
	`  …`
	`  <a:t>Sample text</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	end example]
	
	[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that

Attributes	Description
	properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*] The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.70).
marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).

rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this attribute is omitted, then a value of 0, or left-to-right is implied.
	
	[*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.
	
	Testتجربة ⟵ rtl = 0
	تجربةTest ⟵ rtl = 1
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`
	`<a:t>تجربة</a:t>`
	`</a:r>`

Attributes	Description
	</a:p> <a:p> <a:pPr rtl="1"/> <a:r> <a:rPr> <a:rtl w:val="0"/> </a:rPr> <a:t>Test </a:t> </a:r> <a:r> <a:t>تجربة</a:t> </a:r> </a:p> </p:txBody> *end example*] 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_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.4.15 lvl3pPr (List Level 3 Text Style)

This element specifies all paragraph level text properties for all elements that have the attribute lvl="2". There are a total of 9 level text property elements allowed, levels 0-8. It is recommended that the order in which this and other level property elements are specified be in order of increasing level. That is lvl2pPr should come before lvl3pPr. This allows the lower level properties to take precedence over the higher level ones because they are parsed first.

[*Example*: Consider the following DrawingML code that would specify a paragraph to follow the level style defined in lvl3pPr and thus create a paragraph of text that has no bullets and is right aligned.

```
<p:txBody>
  …
  <a:lstStyle>
    <a:lvl3pPr algn="r">
      <a:buNone/>
    </a:lvl3pPr>
  </a:lstStyle>
```

```
    <a:p>
      <a:pPr lvl="2">
      </a:pPr>
      …
      <a:t>Some text</a:t>
      …
    </a:p>
  </p:txBody>
```

end example]

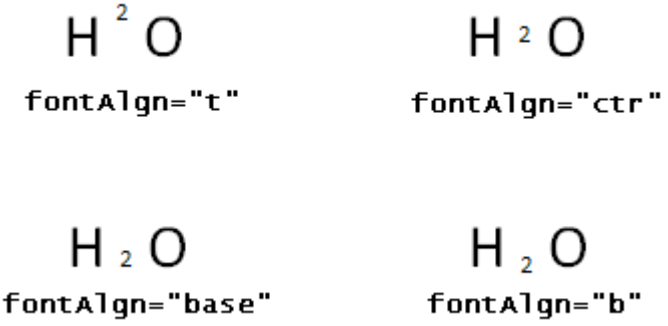
[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]

Attributes	Description
algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied. Sample text [*Example*: Consider the case where the user wishes to have two columns of text that have a justified alignment, much like text within a book. The following DrawingML could describe this. `<p:txBody>` `<a:bodyPr numCol="2" spcCol="914400"…/>` `<a:normAutofit/>` `</a:bodyPr>` `…` `<a:p>` `<a:pPr marL="0" algn="just">`

Attributes	Description
	<a:buNone/> </a:pPr> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextAlignType simple type (§20.1.10.58).
defTabSz (Default Tab Size)	Specifies the default size for a tab character within this paragraph. This attribute should be used to describe the spacing of tabs within the paragraph instead of a leading indentation tab. For indentation tabs there are the marL and indent attributes to assist with this. [*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this. <p:txBody> … <a:p> <a:pPr defTabSz="376300" …/> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the

Attributes	Description

	long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr eaLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied. [*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below. 

Attributes	Description
	<a:txtBody> … <a:pPr fontAlgn="b" …/> … <a:r> <a:rPr …/> <a:t>H </a:t> </a:r> <a:r> <a:rPr sz="1200" …/> <a:t>2</a:t> </a:r> <a:r> <a:rPr …/> <a:t>O</a:t> </a:r> … </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.65).
hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this attribute is omitted, then a value of -342900 is implied.

Attributes	Description

	Here is some Sample text
	
	[*Example*: Consider the scenario where the user now wanted to add a paragraph indentation to the first line of text in their two column format book.
	
	`<p:txBody>`
	`  <a:bodyPr numCol="2" spcCol="914400"…/>`
	`    <a:normAutofit/>`
	`    </a:bodyPr>`
	`  …`
	`  <a:p>`
	`    <a:pPr marL="0" indent="571500" algn="just">`
	`      <a:buNone/>`
	`    </a:pPr>`
	`    …`
	`    <a:t>Here is some…</a:t>`
	`    …`
	`  </a:p>`
	`</p:txBody>`
	
	By adding the indent attribute the user has effectively added a first line indent to this paragraph of text. *end example*]
	
	The possible values for this attribute are defined by the ST_TextIndent simple type (§20.1.10.69).

latinLnBrk (Latin Line Break)	Specifies whether a Latin word can be broken in half and wrapped onto the next line without a hyphen being added. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. It is not present within the existence of normal breakable Latin words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of `1` or `true` is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken

Attributes	Description

	for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr latinLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values.
	
	[*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr lvl="1" …/>`
	`  …`
	`  <a:t>Sample text</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	end example]
	
	[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that

Attributes	Description
	properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*] The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.70).
marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).

rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this attribute is omitted, then a value of 0, or left-to-right is implied.
	
	[*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.
	
	Testتجربة ⟵ rtl = 0
	تجربةTest ⟵ rtl = 1
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`
	`<a:t>تجربة</a:t>`
	`</a:r>`

Attributes	Description
	</a:p> <a:p> <a:pPr rtl="1"/> <a:r> <a:rPr> <a:rtl w:val="0"/> </a:rPr> <a:t>Test </a:t> </a:r> <a:r> <a:t>تجربة</a:t> </a:r> </a:p> </p:txBody> *end example*] 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_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.4.16 lvl4pPr (List Level 4 Text Style)

This element specifies all paragraph level text properties for all elements that have the attribute lvl="3". There are a total of 9 level text property elements allowed, levels 0-8. It is recommended that the order in which this and other level property elements are specified be in order of increasing level. That is lvl2pPr should come before lvl3pPr. This allows the lower level properties to take precedence over the higher level ones because they are parsed first.

[*Example*: Consider the following DrawingML code that would specify a paragraph to follow the level style defined in lvl4pPr and thus create a paragraph of text that has no bullets and is right aligned.

```
<p:txBody>
  …
  <a:lstStyle>
    <a:lvl4pPr algn="r">
      <a:buNone/>
    </a:lvl4pPr>
  </a:lstStyle>
```

```
    <a:p>
      <a:pPr lvl="3">
      </a:pPr>
      …
      <a:t>Some text</a:t>
      …
    </a:p>
  </p:txBody>
```

end example]

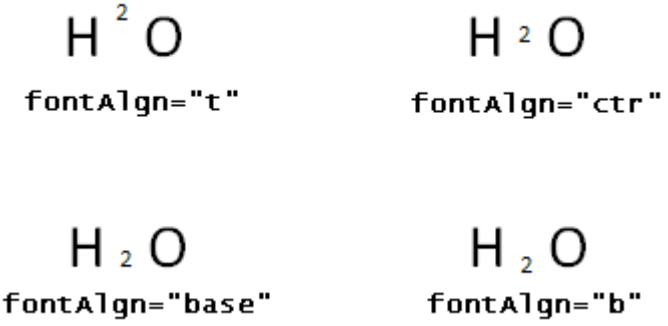
[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]

Attributes	Description
algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied. Sample text [*Example*: Consider the case where the user wishes to have two columns of text that have a justified alignment, much like text within a book. The following DrawingML could describe this. `<p:txBody>` `<a:bodyPr numCol="2" spcCol="914400"…/>` `<a:normAutofit/>` `</a:bodyPr>` `…` `<a:p>` `<a:pPr marL="0" algn="just">`

Attributes	Description
	<a:buNone/> </a:pPr> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextAlignType simple type (§20.1.10.58).
defTabSz (Default Tab Size)	Specifies the default size for a tab character within this paragraph. This attribute should be used to describe the spacing of tabs within the paragraph instead of a leading indentation tab. For indentation tabs there are the marL and indent attributes to assist with this. [*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this. <p:txBody> … <a:p> <a:pPr defTabSz="376300" …/> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the

Attributes	Description

	long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr eaLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied. [*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below. 

Attributes	Description
	<a:txtBody> … <a:pPr fontAlgn="b" …/> … <a:r> <a:rPr …/> <a:t>H </a:t> </a:r> <a:r> <a:rPr sz="1200" …/> <a:t>2</a:t> </a:r> <a:r> <a:rPr …/> <a:t>O</a:t> </a:r> … </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.65).
hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this attribute is omitted, then a value of -342900 is implied.

Attributes	Description

	Here is some Sample text
	
	[*Example*: Consider the scenario where the user now wanted to add a paragraph indentation to the first line of text in their two column format book.
	
	`<p:txBody>`
	`  <a:bodyPr numCol="2" spcCol="914400"…/>`
	`    <a:normAutofit/>`
	`    </a:bodyPr>`
	`  …`
	`  <a:p>`
	`    <a:pPr marL="0" indent="571500" algn="just">`
	`      <a:buNone/>`
	`    </a:pPr>`
	`    …`
	`    <a:t>Here is some…</a:t>`
	`    …`
	`  </a:p>`
	`</p:txBody>`
	
	By adding the indent attribute the user has effectively added a first line indent to this paragraph of text. *end example*]
	
	The possible values for this attribute are defined by the ST_TextIndent simple type (§20.1.10.69).

latinLnBrk (Latin Line Break)	Specifies whether a Latin word can be broken in half and wrapped onto the next line without a hyphen being added. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. It is not present within the existence of normal breakable Latin words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of `1` or `true` is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken

Attributes	Description

	for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr latinLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values.
	
	[*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr lvl="1" …/>`
	`  …`
	`  <a:t>Sample text</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	end example]
	
	[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that

Attributes	Description
	properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*] The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.70).
marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).

rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this attribute is omitted, then a value of 0, or left-to-right is implied.
	
	[*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.
	
	Testتجربة ⟵ rtl = 0
	تجربةTest ⟵ rtl = 1
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`
	`<a:t>تجربة</a:t>`
	`</a:r>`

Attributes	Description
	</a:p> <a:p> <a:pPr rtl="1"/> <a:r> <a:rPr> <a:rtl w:val="0"/> </a:rPr> <a:t>Test </a:t> </a:r> <a:r> <a:t>تجربة</a:t> </a:r> </a:p> </p:txBody> *end example*] 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_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.4.17 lvl5pPr (List Level 5 Text Style)

This element specifies all paragraph level text properties for all elements that have the attribute lvl="4". There are a total of 9 level text property elements allowed, levels 0-8. It is recommended that the order in which this and other level property elements are specified be in order of increasing level. That is lvl2pPr should come before lvl3pPr. This allows the lower level properties to take precedence over the higher level ones because they are parsed first.

[*Example*: Consider the following DrawingML code that would specify a paragraph to follow the level style defined in lvl5pPr and thus create a paragraph of text that has no bullets and is right aligned.

```
<p:txBody>
  …
  <a:lstStyle>
    <a:lvl5pPr algn="r">
      <a:buNone/>
    </a:lvl5pPr>
  </a:lstStyle>
```

```
    <a:p>
      <a:pPr lvl="4">
      </a:pPr>
      …
      <a:t>Some text</a:t>
      …
    </a:p>
  </p:txBody>
```

end example]

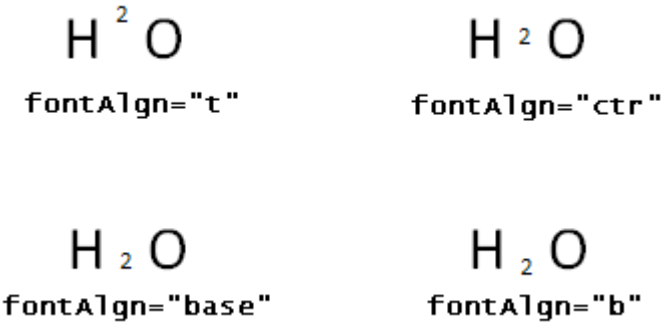
[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]

Attributes	Description
algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied. Sample text [*Example*: Consider the case where the user wishes to have two columns of text that have a justified alignment, much like text within a book. The following DrawingML could describe this. `<p:txBody>` `  <a:bodyPr numCol="2" spcCol="914400"…/>` `  <a:normAutofit/>` `  </a:bodyPr>` `  …` `  <a:p>` `  <a:pPr marL="0" algn="just">`

Attributes	Description
	<a:buNone/> </a:pPr> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextAlignType simple type (§20.1.10.58).
defTabSz (Default Tab Size)	Specifies the default size for a tab character within this paragraph. This attribute should be used to describe the spacing of tabs within the paragraph instead of a leading indentation tab. For indentation tabs there are the marL and indent attributes to assist with this. [*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this. <p:txBody> … <a:p> <a:pPr defTabSz="376300" …/> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the

Attributes	Description

	long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr eaLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied. [*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below. 

Attributes	Description
	<a:txtBody> … <a:pPr fontAlgn="b" …/> … <a:r> <a:rPr …/> <a:t>H </a:t> </a:r> <a:r> <a:rPr sz="1200" …/> <a:t>2</a:t> </a:r> <a:r> <a:rPr …/> <a:t>O</a:t> </a:r> … </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.65).
hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this attribute is omitted, then a value of -342900 is implied.

Attributes	Description

	Here is some Sample text
	
	[*Example*: Consider the scenario where the user now wanted to add a paragraph indentation to the first line of text in their two column format book.
	
	`<p:txBody>`
	`  <a:bodyPr numCol="2" spcCol="914400"…/>`
	`    <a:normAutofit/>`
	`    </a:bodyPr>`
	`  …`
	`  <a:p>`
	`    <a:pPr marL="0" indent="571500" algn="just">`
	`      <a:buNone/>`
	`    </a:pPr>`
	`    …`
	`    <a:t>Here is some…</a:t>`
	`    …`
	`  </a:p>`
	`</p:txBody>`
	
	By adding the indent attribute the user has effectively added a first line indent to this paragraph of text. *end example*]
	
	The possible values for this attribute are defined by the ST_TextIndent simple type (§20.1.10.69).

latinLnBrk (Latin Line Break)	Specifies whether a Latin word can be broken in half and wrapped onto the next line without a hyphen being added. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. It is not present within the existence of normal breakable Latin words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or `true` is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken

Attributes	Description

	for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr latinLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values.
	
	[*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr lvl="1" …/>`
	`  …`
	`  <a:t>Sample text</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	end example]
	
	[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that

Attributes	Description
	properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*] The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.70).
marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).

rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this attribute is omitted, then a value of 0, or left-to-right is implied.
	
	[*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.
	
	Testتجربة ⟵ rtl = 0
	تجربةTest ⟵ rtl = 1
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`
	`<a:t>تجربة</a:t>`
	`</a:r>`

Attributes	Description
	</a:p> <a:p> <a:pPr rtl="1"/> <a:r> <a:rPr> <a:rtl w:val="0"/> </a:rPr> <a:t>Test </a:t> </a:r> <a:r> <a:t>تجربة</a:t> </a:r> </a:p> </p:txBody> *end example*] 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_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.4.18 lvl6pPr (List Level 6 Text Style)

This element specifies all paragraph level text properties for all elements that have the attribute lvl="5". There are a total of 9 level text property elements allowed, levels 0-8. It is recommended that the order in which this and other level property elements are specified be in order of increasing level. That is lvl2pPr should come before lvl3pPr. This allows the lower level properties to take precedence over the higher level ones because they are parsed first.

[*Example*: Consider the following DrawingML code that would specify a paragraph to follow the level style defined in lvl6pPr and thus create a paragraph of text that has no bullets and is right aligned.

```
<p:txBody>
  …
  <a:lstStyle>
    <a:lvl6pPr algn="r">
      <a:buNone/>
    </a:lvl6pPr>
  </a:lstStyle>
```

```
    <a:p>
      <a:pPr lvl="5">
      </a:pPr>
      …
      <a:t>Some text</a:t>
      …
    </a:p>
  </p:txBody>
```

end example]

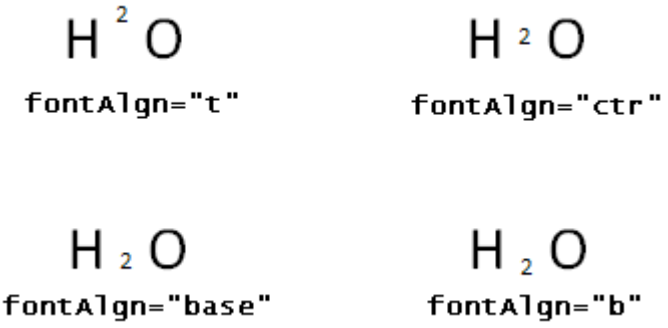
[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]

Attributes	Description
algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied. Sample text [*Example*: Consider the case where the user wishes to have two columns of text that have a justified alignment, much like text within a book. The following DrawingML could describe this. `<p:txBody>` `  <a:bodyPr numCol="2" spcCol="914400"…/>` `    <a:normAutofit/>` `  </a:bodyPr>` `  …` `  <a:p>` `    <a:pPr marL="0" algn="just">`

Attributes	Description
	<a:buNone/> </a:pPr> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextAlignType simple type (§20.1.10.58).
defTabSz (Default Tab Size)	Specifies the default size for a tab character within this paragraph. This attribute should be used to describe the spacing of tabs within the paragraph instead of a leading indentation tab. For indentation tabs there are the marL and indent attributes to assist with this. [*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this. <p:txBody> … <a:p> <a:pPr defTabSz="376300" …/> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the

Attributes	Description

	long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr eaLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied. [*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below. 

Attributes	Description
	<a:txtBody> … <a:pPr fontAlgn="b" …/> … <a:r> <a:rPr …/> <a:t>H </a:t> </a:r> <a:r> <a:rPr sz="1200" …/> <a:t>2</a:t> </a:r> <a:r> <a:rPr …/> <a:t>O</a:t> </a:r> … </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.65).
hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this attribute is omitted, then a value of -342900 is implied.

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>Here is some Sample text

[Example: Consider the scenario where the user now wanted to add a paragraph indentation to the first line of text in their two column format book.

<pre><p:txBody>
  <a:bodyPr numCol="2" spcCol="914400"…/>
    <a:normAutofit/>
    </a:bodyPr>
  …
  <a:p>
    <a:pPr marL="0" indent="571500" algn="just">
      <a:buNone/>
    </a:pPr>
    …
    <a:t>Here is some…</a:t>
    …
  </a:p>
</p:txBody></pre>By adding the indent attribute the user has effectively added a first line indent to this paragraph of text. end example]

The possible values for this attribute are defined by the ST_TextIndent simple type (§20.1.10.69).</td></tr>
<tr><td>latinLnBrk (Latin Line Break)</td><td>Specifies whether a Latin word can be broken in half and wrapped onto the next line without a hyphen being added. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. It is not present within the existence of normal breakable Latin words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied.

[Example: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken</td></tr>
</table>

Attributes	Description

	for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr latinLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values.
	
	[*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr lvl="1" …/>`
	`  …`
	`  <a:t>Sample text</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	end example]
	
	[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that

Attributes	Description
	properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*] The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.70).
marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).

rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this attribute is omitted, then a value of 0, or left-to-right is implied.
	
	[*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.
	
	Testتجربة ⟵ rtl = 0
	تجربةTest ⟵ rtl = 1
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`
	`<a:t>تجربة</a:t>`
	`</a:r>`

Attributes	Description
	</a:p> <a:p> <a:pPr rtl="1"/> <a:r> <a:rPr> <a:rtl w:val="0"/> </a:rPr> <a:t>Test </a:t> </a:r> <a:r> <a:t>تجربة</a:t> </a:r> </a:p> </p:txBody> *end example*] 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_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.4.19 lvl7pPr (List Level 7 Text Style)

This element specifies all paragraph level text properties for all elements that have the attribute lvl="6". There are a total of 9 level text property elements allowed, levels 0-8. It is recommended that the order in which this and other level property elements are specified be in order of increasing level. That is lvl2pPr should come before lvl3pPr. This allows the lower level properties to take precedence over the higher level ones because they are parsed first.

[*Example*: Consider the following DrawingML code that would specify a paragraph to follow the level style defined in lvl7pPr and thus create a paragraph of text that has no bullets and is right aligned.

```
<p:txBody>
  …
  <a:lstStyle>
    <a:lvl7pPr algn="r">
      <a:buNone/>
    </a:lvl7pPr>
  </a:lstStyle>
```

```
    <a:p>
      <a:pPr lvl="6">
      </a:pPr>
      …
      <a:t>Some text</a:t>
      …
    </a:p>
  </p:txBody>
```

end example]

[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]

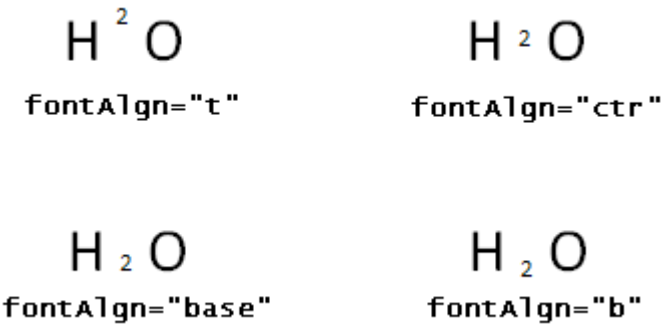
Attributes	Description

algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied.
	
	Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text
	
	Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text
	
	[*Example*: Consider the case where the user wishes to have two columns of text that have a justified alignment, much like text within a book. The following DrawingML could describe this.
	
	`<p:txBody>`
	`  <a:bodyPr numCol="2" spcCol="914400"…/>`
	`    <a:normAutofit/>`
	`  </a:bodyPr>`
	`  …`
	`  <a:p>`
	`    <a:pPr marL="0" algn="just">`

Attributes	Description
	<a:buNone/> </a:pPr> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextAlignType simple type (§20.1.10.58).
defTabSz (Default Tab Size)	Specifies the default size for a tab character within this paragraph. This attribute should be used to describe the spacing of tabs within the paragraph instead of a leading indentation tab. For indentation tabs there are the marL and indent attributes to assist with this. [*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this. <p:txBody> … <a:p> <a:pPr defTabSz="376300" …/> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the

Attributes	Description

	long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr eaLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied. [*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below. 

Attributes	Description
	<a:txtBody> … <a:pPr fontAlgn="b" …/> … <a:r> <a:rPr …/> <a:t>H </a:t> </a:r> <a:r> <a:rPr sz="1200" …/> <a:t>2</a:t> </a:r> <a:r> <a:rPr …/> <a:t>O</a:t> </a:r> … </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.65).
hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this attribute is omitted, then a value of -342900 is implied.

Attributes	Description

	Here is some Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text
	
	[*Example*: Consider the scenario where the user now wanted to add a paragraph indentation to the first line of text in their two column format book.
	
	<p:txBody>
	  <a:bodyPr numCol="2" spcCol="914400"…/>
	    <a:normAutofit/>
	    </a:bodyPr>
	  …
	  <a:p>
	    <a:pPr marL="0" indent="571500" algn="just">
	      <a:buNone/>
	    </a:pPr>
	    …
	    <a:t>Here is some…</a:t>
	    …
	  </a:p>
	</p:txBody>
	
	By adding the indent attribute the user has effectively added a first line indent to this paragraph of text. *end example*]
	
	The possible values for this attribute are defined by the ST_TextIndent simple type (§20.1.10.69).

latinLnBrk (Latin Line Break)	Specifies whether a Latin word can be broken in half and wrapped onto the next line without a hyphen being added. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. It is not present within the existence of normal breakable Latin words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of `1` or `true` is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken

Attributes	Description

	for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr latinLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values.
	
	[*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr lvl="1" …/>`
	`  …`
	`  <a:t>Sample text</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	end example]
	
	[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that

Attributes	Description
	properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*] The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.70).
marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).

rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this attribute is omitted, then a value of 0, or left-to-right is implied.
	
	[*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.
	
	Testتجربة ⟵ rtl = 0
	تجربةTest ⟵ rtl = 1
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`
	`<a:t>تجربة</a:t>`
	`</a:r>`

Attributes	Description
	</a:p> <a:p> <a:pPr rtl="1"/> <a:r> <a:rPr> <a:rtl w:val="0"/> </a:rPr> <a:t>Test </a:t> </a:r> <a:r> <a:t>تجربة</a:t> </a:r> </a:p> </p:txBody> *end example*] 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_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.4.20 lvl8pPr (List Level 8 Text Style)

This element specifies all paragraph level text properties for all elements that have the attribute lvl="7". There are a total of 9 level text property elements allowed, levels 0-8. It is recommended that the order in which this and other level property elements are specified be in order of increasing level. That is lvl2pPr should come before lvl3pPr. This allows the lower level properties to take precedence over the higher level ones because they are parsed first.

[*Example*: Consider the following DrawingML code that would specify a paragraph to follow the level style defined in lvl8pPr and thus create a paragraph of text that has no bullets and is right aligned.

```
<p:txBody>
  …
  <a:lstStyle>
    <a:lvl8pPr algn="r">
      <a:buNone/>
    </a:lvl8pPr>
  </a:lstStyle>
  <a:p>
    <a:pPr lvl="7">
    </a:pPr>
    …
```

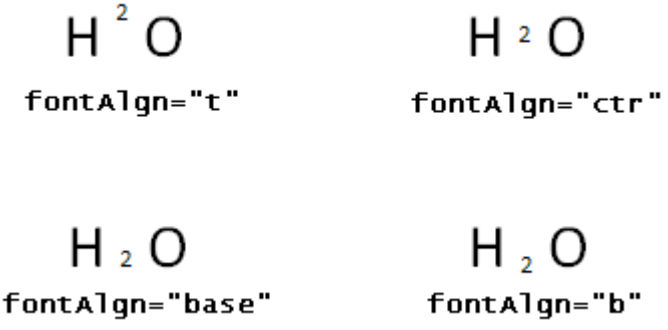
```
      <a:t>Some text</a:t>
      …
    </a:p>
  </p:txBody>
```

end example]

[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]

Attributes	Description
algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied. Sample text [*Example*: Consider the case where the user wishes to have two columns of text that have a justified alignment, much like text within a book. The following DrawingML could describe this. `<p:txBody>` `<a:bodyPr numCol="2" spcCol="914400"…/>` `<a:normAutofit/>` `</a:bodyPr>` `…` `<a:p>` `<a:pPr marL="0" algn="just">` `<a:buNone/>` `</a:pPr>` `…` `<a:t>Sample Text …</a:t>`

Attributes	Description
	… </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextAlignType simple type (§20.1.10.58).
defTabSz (Default Tab Size)	Specifies the default size for a tab character within this paragraph. This attribute should be used to describe the spacing of tabs within the paragraph instead of a leading indentation tab. For indentation tabs there are the marL and indent attributes to assist with this. [*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this. <p:txBody> … <a:p> <a:pPr defTabSz="376300" …/> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows. <p:txBody> …

Attributes	Description
	<a:p> <a:pPr eaLnBrk="0" …/> … <a:t>Sample text (Long word)</a:t> … </a:p> </p:txBody> Sample text Sample text Sample text supercalifr agilisticexpialidocious Sample text Sample text Sample text supercalifragilisticexpialidocious *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied. [*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below.  <a:txtBody> … <a:pPr fontAlgn="b" …/>

Attributes	Description

	`…`
	`  <a:r>`
	`    <a:rPr …/>`
	`    <a:t>H </a:t>`
	`  </a:r>`
	`  <a:r>`
	`    <a:rPr sz="1200" …/>`
	`      <a:t>2</a:t>`
	`  </a:r>`
	`  <a:r>`
	`    <a:rPr …/>`
	`      <a:t>O</a:t>`
	`  </a:r>`
	`  …`
	`</p:txBody>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.65).

hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this attribute is omitted, then a value of -342900 is implied. Here is some Sample text

Attributes	Description

	[*Example*: Consider the scenario where the user now wanted to add a paragraph indentation to the first line of text in their two column format book.
	
	`<p:txBody>`
	`  <a:bodyPr numCol="2" spcCol="914400"…/>`
	`  <a:normAutofit/>`
	`  </a:bodyPr>`
	`  …`
	`  <a:p>`
	`  <a:pPr marL="0" indent="571500" algn="just">`
	`  <a:buNone/>`
	`  </a:pPr>`
	`  …`
	`  <a:t>Here is some…</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	By adding the indent attribute the user has effectively added a first line indent to this paragraph of text. *end example*]
	
	The possible values for this attribute are defined by the ST_TextIndent simple type (§20.1.10.69).

latinLnBrk (Latin Line Break)	Specifies whether a Latin word can be broken in half and wrapped onto the next line without a hyphen being added. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. It is not present within the existence of normal breakable Latin words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of `1` or `true` is implied.
	
	[*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr latinLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`

Attributes	Description
	Sample text Sample text Sample text supercalifr agilisticexpialidocious Sample text Sample text Sample text supercalifragilisticexpialidocious *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values.
	
	[*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr lvl="1" …/>`
	`  …`
	`  <a:t>Sample text</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	end example]
	
	[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]
	
	The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.70).

marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied.

Attributes	Description
	The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).

rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this attribute is omitted, then a value of 0, or left-to-right is implied.
	
	[*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.
	
	 
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`
	`<a:t>تجربة</a:t>`
	`</a:r>`
	`</a:p>`
	`<a:p>`
	`<a:pPr rtl="1"/>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="0"/>`
	`</a:rPr>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:t>تجربة </a:t>`
	`</a:r>`
	`</a:p>`

Attributes	Description
	</p:txBody> *end example*] 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_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.4.21 lvl9pPr (List Level 9 Text Style)

This element specifies all paragraph level text properties for all elements that have the attribute lvl="8". There are a total of 9 level text property elements allowed, levels 0-8. It is recommended that the order in which this and other level property elements are specified be in order of increasing level. That is lvl2pPr should come before lvl3pPr. This allows the lower level properties to take precedence over the higher level ones because they are parsed first.

[*Example*: Consider the following DrawingML code that would specify a paragraph to follow the level style defined in lvl9pPr and thus create a paragraph of text that has no bullets and is right aligned.

```
<p:txBody>
  …
  <a:lstStyle>
    <a:lvl9pPr algn="r">
      <a:buNone/>
    </a:lvl9pPr>
  </a:lstStyle>
  <a:p>
    <a:pPr lvl="8">
    </a:pPr>
    …
    <a:t>Some text</a:t>
    …
  </a:p>
</p:txBody>
```

end example]

[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr

elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>algn (Alignment)</td><td>Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied.

Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text
Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text Sample text

[Example: Consider the case where the user wishes to have two columns of text that have a justified alignment, much like text within a book. The following DrawingML could describe this.

<p:txBody>
  <a:bodyPr numCol="2" spcCol="914400"…/>
    <a:normAutofit/>
  </a:bodyPr>
  …
  <a:p>
    <a:pPr marL="0" algn="just">
      <a:buNone/>
    </a:pPr>
    …
    <a:t>Sample Text …</a:t>
    …
  </a:p>
</p:txBody>

end example]

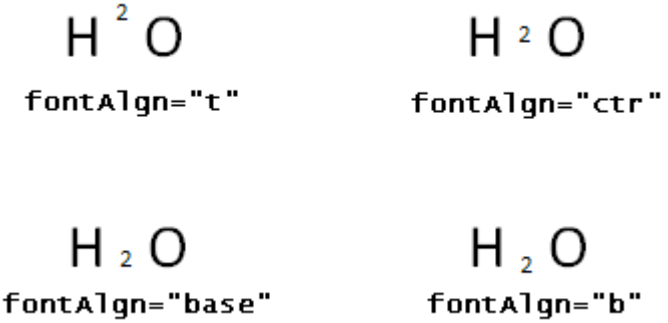
The possible values for this attribute are defined by the ST_TextAlignType simple type (§20.1.10.58).</td></tr>
<tr><td>defTabSz (Default Tab Size)</td><td>Specifies the default size for a tab character within this paragraph. This attribute should be used to describe the spacing of tabs within the paragraph instead of a leading indentation tab. For indentation tabs there are the marL and indent attributes to assist</td></tr>
</table>

Attributes	Description

	with this.
	
	[*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr defTabSz="376300" …/>`
	`  …`
	`  <a:t>Sample Text …</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).

eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of `1` or `true` is implied.
	
	[*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr eaLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr
	agilisticexpialidocious

Attributes	Description
	Sample text Sample text Sample text supercalifragilisticexpialidocious *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied.
	
	[*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below.
	
	 
	
	`<a:txtBody>`
	`…`
	`<a:pPr fontAlgn="b" …/>`
	`…`
	`<a:r>`
	`<a:rPr …/>`
	`<a:t>H </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr sz="1200" …/>`
	`<a:t>2</a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr …/>`
	`<a:t>O</a:t>`

Attributes	Description
	</a:r> … </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.65).
hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this attribute is omitted, then a value of -342900 is implied. Here is some Sample text [*Example*: Consider the scenario where the user now wanted to add a paragraph indentation to the first line of text in their two column format book. <p:txBody> <a:bodyPr numCol="2" spcCol="914400"…/> <a:normAutofit/> </a:bodyPr> … <a:p> <a:pPr marL="0" indent="571500" algn="just"> <a:buNone/>

Attributes	Description
	</a:pPr> … <a:t>Here is some…</a:t> … </a:p> </p:txBody> By adding the indent attribute the user has effectively added a first line indent to this paragraph of text. *end example*] The possible values for this attribute are defined by the ST_TextIndent simple type (§20.1.10.69).
latinLnBrk (Latin Line Break)	Specifies whether a Latin word can be broken in half and wrapped onto the next line without a hyphen being added. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. It is not present within the existence of normal breakable Latin words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows. <p:txBody> … <a:p> <a:pPr latinLnBrk="0" …/> … <a:t>Sample text (Long word)</a:t> … </a:p> </p:txBody> Sample text Sample text Sample text supercalifr agilisticexpialidocious Sample text Sample text Sample text supercalifragilisticexpialidocious *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this

Attributes	Description

	attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values.
	
	[*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr lvl="1" …/>`
	`  …`
	`  <a:t>Sample text</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	end example]
	
	[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]
	
	The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.70).

marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.71).
rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this

Attributes	Description

	attribute is omitted, then a value of 0, or left-to-right is implied.
	
	[*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.
	
	 
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`
	`<a:t>تجربة</a:t>`
	`</a:r>`
	`</a:p>`
	`<a:p>`
	`<a:pPr rtl="1"/>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="0"/>`
	`</a:rPr>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:t>تجربة </a:t>`
	`</a:r>`
	`</a:p>`
	`</p:txBody>`
	
	end example]
	
	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_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.5 Font Substitution

If any DrawingML element references a font and an appropriate format of the font is not stored within the document, the process of finding a suitable alternative font is known as *font substitution*.

The following elements specify font information: buFont@panose (§21.1.2.4.6); cs@panose (§21.1.2.3.1); ea@panose (§21.1.2.3.3); font@panose (§**19.2.1.13**); latin@panose (§21.1.2.3.7); sym@panose (§21.1.2.3.10).

The exact algorithm which is used for font substitution is highly dependent on the characteristics which are most desirable when performing the substitution: similar appearance of each glyph (to maximize visual familiarity), similar physical characteristics (to minimize changes in line height and breaking), etc. ISO/IEC 29500 recommends that applications look for the closest match to the following attribute values (in descending priority) in order to determine a suitable alternative font: panose, charset, pitchFamily, and typeface; however, applications are free to apply higher-order logic in its place.

21.1.3 Tables

This section contains information regarding the definition of a table within DrawingML. The following image is an example table within DrawingML.

text	text	text	text	text
text	text	text	text	text
text	text	text	text	text
text	text	text	text	text
text	text	text	text	text

21.1.3.1 cell3D (Cell 3-D)

This element specifies a set of properties which dictate the 3-D appearance of a given cell in a table. Collectively, these properties are referred to as a cell 3-D. The application of these properties occurs on a per-cell basis in the table.

Attributes	Description
prstMaterial (Preset Material)	Specifies a material type which is used to define the material characteristics of the cell. The material properties, combined with the lighting characteristics of the scene in define the final look and feel of the 3-D appearance of the cell. The possible values for this attribute are defined by the ST_PresetMaterialType simple

Attributes	Description
	type (§20.1.10.49).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Cell3D) is located in §A.4.1. *end note*]

21.1.3.2 gridCol (Table Grid Column)

This element specifies the width of a given column within a table. For each column in a table, there is an associated table grid column defining the width of the column.

[*Example*: Consider the following example of a table grid containing widths defined for three table grid columns:

```
<a:tblGrid>
  <a:gridCol w="1117600"/>
  <a:gridCol w="1117600"/>
  <a:gridCol w="1117600"/>
</a:tblGrid>
```

end example]

Attributes	Description
w (Width)	The width of the column in EMUs. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableCol) is located in §A.4.1. *end note*]

21.1.3.3 header (Header Cell Reference)

This element specifies a reference, using a unique identifier, to a table header cell that is associated with the current table cell. The identifier representing the reference shall be stored on this element's val attribute and is used to reference the unique identifier value of a table header cell. The contents of the table header cell designated by a specific unique identifier shall be used as the table header information associated with the table cell that references that specific unique identifier.

If this element is omitted or the value of the header cell identifier cannot be resolved, no header cell shall be associated with the current table cell.

[*Example*: Consider the following 3 x 3 table with four header cells with values A, B, C, and D and four data cells with values x1, x2, y1, and y2:

	A	B
C	x1	x2
D	y1	y2

Each of the data cells is associated with two header cells and can be represented in DrawingML as follows:

```
<a:tbl>
  …
  <a:tr>
    …
    <a:tc >
      …
    </a:tc>
    <a:tc id="HeaderA">
      …
      <a:p>
        <a:r>
          <a:t>A</a:t>
        </a:r>
      </a:p>
    </a:tc>
    …
  </a:tr>
  <a:tr>
    …
    <a:tc id="HeaderC">
      …
      <a:p>
        <a:r>
          <a:t>C</a:t>
        </a:r>
      </a:p>
    </a:tc>
    <a:tc>

      <a:p>
        <a:r>
          <a:t>x1</a:t>
        </a:r>
      </a:p>
      <a:tcPr>
        …
        <a:headers>
          <a:header val="HeaderA" />
```

```
            <a:header val="HeaderC" />
          </a:headers>
          …
        </a:tcPr>
      </a:tc>
      …
    </a:tr>
  </a:tbl>
```

The headers element specifies the list of header cells associated with the table cell that has a value of x1. In this example the table cell with the content value of x1 is associated with headers that have an id of HeaderA and HeaderC. *end example*]

The possible values for this element are defined by the W3C XML Schema string datatype.

21.1.3.4 headers (Header Cells Associated With Table Cell)

This element specifies the list of header cells, as specified by children header elements, that provide header information associated with the current table cell. Each header cell shall specify a unique identifier, as specified by the use of the attribute id on the header cell tc element. This element is typically used to gather header information about data and sub header cells.

If this element is omitted or there exists no children header element, no header cell shall be associated with the given table cell.

[*Example*: Consider the following 3 x 3 table with four header cells with values A, B, C, and D and four data cells with values x1, x2, y1, and y2:

	A	B
C	x1	x2
D	y1	y2

Each of the data cells is associated with two header cells and can be represented in DrawingML as follows:

```
<a:tbl>
  …
  <a:tr>
    …
    <a:tc >
      …
    </a:tc>
    <a:tc id="HeaderA">
      …
      <a:p>
        <a:r>
          <a:t>A</a:t>
```



```
          </a:r>
        </a:p>
      </a:tc>
      …
    </a:tr>
    <a:tr>
      …
      <a:tc id="HeaderC">
        …
        <a:p>
          <a:r>
            <a:t>C</a:t>
          </a:r>
        </a:p>
      </a:tc>
      <a:tc>
        <a:p>
          <a:r>
            <a:t>x1</a:t>
          </a:r>
        </a:p>
        <a:tcPr>
          …
          <a:headers>
            <a:header val="HeaderA" />
            <a:header val="HeaderC" />
          </a:headers>
          …
        </a:tcPr>
      </a:tc>
      …
    </a:tr>
  </a:tbl>
```

The headers element specifies the list of header cells associated with the table cell that has a value of x1. In this example x1 is associated with headers that have an id of HeaderA and HeaderC. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Headers) is located in §A.4.1. *end note*]

21.1.3.5 lnB (Bottom Border Line Properties)

This element defines the line properties associated with the bottom border of a given cell.

[*Example*: Consider the following example of a bottom border line properties element within DrawingML:


```
<a:lnB w="38100" cap="flat" cmpd="sng" algn="ctr">
  <a:solidFill>
    <a:schemeClr val="accent2"/>
  </a:solidFill>
  <a:prstDash val="solid"/>
  <a:round/>
  <a:headEnd type="none" w="med" len="med"/>
  <a:tailEnd type="none" w="med" len="med"/>
</a:lnB>
```

In this example, one can see that the bottom border line style defined with certain properties, such as a flat end line cap, a given width, head and tail end, color, etc. *end example*]

Attributes	Description
algn (Stroke Alignment)	Specifies the alignment to be used for the underline stroke. The possible values for this attribute are defined by the ST_PenAlignment simple type (§20.1.10.39).
cap (Line Ending Cap Type)	Specifies the ending caps that should be used for this line. [*Note*: Examples of cap types are rounded, flat, etc. *end note*] If this attribute is omitted, than a value of square is assumed. The possible values for this attribute are defined by the ST_LineCap simple type (§20.1.10.31).
cmpd (Compound Line Type)	Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. The possible values for this attribute are defined by the ST_CompoundLine simple type (§20.1.10.15).
w (Line Width)	Specifies the width to be used for the underline stroke. If this attribute is omitted, then a value of 0 is assumed. The possible values for this attribute are defined by the ST_LineWidth simple type (§20.1.10.35).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineProperties) is located in §A.4.1. *end note*]

21.1.3.6 lnBlToTr (Bottom-Left to Top-Right Border Line Properties)

This element defines the line properties associated with the diagonal line from the bottom left corner of the cell to the top right corner.

[*Example*: Consider the following example of a lnBlToTr within DrawingML:


```
<a:lnBlToTr w="38100" cap="flat" cmpd="sng" algn="ctr">
  <a:solidFill>
    <a:schemeClr val="accent2"/>
  </a:solidFill>
  <a:prstDash val="solid"/>
  <a:round/>
  <a:headEnd type="none" w="med" len="med"/>
  <a:tailEnd type="none" w="med" len="med"/>
</a:lnBlToTr >
```

In this example, one can see that the border line style defined with certain properties, such as a flat end line cap, a given width, head and tail end, color, etc. *end example*]

Attributes	Description
algn (Stroke Alignment)	Specifies the alignment to be used for the underline stroke. The possible values for this attribute are defined by the ST_PenAlignment simple type (§20.1.10.39).
cap (Line Ending Cap Type)	Specifies the ending caps that should be used for this line. [*Note*: Examples of cap types are rounded, flat, etc. *end note*] If this attribute is omitted, than a value of square is assumed. The possible values for this attribute are defined by the ST_LineCap simple type (§20.1.10.31).
cmpd (Compound Line Type)	Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. The possible values for this attribute are defined by the ST_CompoundLine simple type (§20.1.10.15).
w (Line Width)	Specifies the width to be used for the underline stroke. If this attribute is omitted, then a value of 0 is assumed. The possible values for this attribute are defined by the ST_LineWidth simple type (§20.1.10.35).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineProperties) is located in §A.4.1. *end note*]

21.1.3.7 lnL (Left Border Line Properties)

This element defines the line properties associated with the left border of a cell

[*Example*: Consider the following example of a lnL within DrawingML:

```
<a:lnL w="38100" cap="flat" cmpd="sng" algn="ctr">
  <a:solidFill>
    <a:schemeClr val="accent2"/>
  </a:solidFill>
  <a:prstDash val="solid"/>
  <a:round/>
  <a:headEnd type="none" w="med" len="med"/>
  <a:tailEnd type="none" w="med" len="med"/>
</a:lnL >
```

In this example, one can see that the border line style defined with certain properties, such as a flat end line cap, a given width, head and tail end, color, etc. *end example*]

Attributes	Description
algn (Stroke Alignment)	Specifies the alignment to be used for the underline stroke. The possible values for this attribute are defined by the ST_PenAlignment simple type (§20.1.10.39).
cap (Line Ending Cap Type)	Specifies the ending caps that should be used for this line. [*Note*: Examples of cap types are rounded, flat, etc. *end note*] If this attribute is omitted, than a value of square is assumed. The possible values for this attribute are defined by the ST_LineCap simple type (§20.1.10.31).
cmpd (Compound Line Type)	Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. The possible values for this attribute are defined by the ST_CompoundLine simple type (§20.1.10.15).
w (Line Width)	Specifies the width to be used for the underline stroke. If this attribute is omitted, then a value of 0 is assumed. The possible values for this attribute are defined by the ST_LineWidth simple type (§20.1.10.35).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineProperties) is located in §A.4.1. *end note*]

21.1.3.8 lnR (Right Border Line Properties)

This element defines the line properties associated with right border of a cell.

[*Example*: Consider the following example of a lnR within DrawingML:


```
<a:lnR w="38100" cap="flat" cmpd="sng" algn="ctr">
  <a:solidFill>
    <a:schemeClr val="accent2"/>
  </a:solidFill>
  <a:prstDash val="solid"/>
  <a:round/>
  <a:headEnd type="none" w="med" len="med"/>
  <a:tailEnd type="none" w="med" len="med"/>
</a:lnR >
```

In this example, one can see that the border line style defined with certain properties, such as a flat end line cap, a given width, head and tail end, color, etc. *end example*]

Attributes	Description
algn (Stroke Alignment)	Specifies the alignment to be used for the underline stroke. The possible values for this attribute are defined by the ST_PenAlignment simple type (§20.1.10.39).
cap (Line Ending Cap Type)	Specifies the ending caps that should be used for this line. [*Note*: Examples of cap types are rounded, flat, etc. *end note*] If this attribute is omitted, than a value of square is assumed. The possible values for this attribute are defined by the ST_LineCap simple type (§20.1.10.31).
cmpd (Compound Line Type)	Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. The possible values for this attribute are defined by the ST_CompoundLine simple type (§20.1.10.15).
w (Line Width)	Specifies the width to be used for the underline stroke. If this attribute is omitted, then a value of 0 is assumed. The possible values for this attribute are defined by the ST_LineWidth simple type (§20.1.10.35).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineProperties) is located in §A.4.1. *end note*]

21.1.3.9 lnT (Top Border Line Properties)

This element defines the line properties associated with the top border of a cell.

[*Example*: Consider the following example of a lnT within DrawingML:

```
<a:lnT w="38100" cap="flat" cmpd="sng" algn="ctr">
  <a:solidFill>
    <a:schemeClr val="accent2"/>
  </a:solidFill>
  <a:prstDash val="solid"/>
  <a:round/>
  <a:headEnd type="none" w="med" len="med"/>
  <a:tailEnd type="none" w="med" len="med"/>
</a:lnT >
```

In this example, one can see that the border line style defined with certain properties, such as a flat end line cap, a given width, head and tail end, color, etc. *end example*]

Attributes	Description
algn (Stroke Alignment)	Specifies the alignment to be used for the underline stroke. The possible values for this attribute are defined by the ST_PenAlignment simple type (§20.1.10.39).
cap (Line Ending Cap Type)	Specifies the ending caps that should be used for this line. [*Note*: Examples of cap types are rounded, flat, etc. *end note*] If this attribute is omitted, than a value of square is assumed. The possible values for this attribute are defined by the ST_LineCap simple type (§20.1.10.31).
cmpd (Compound Line Type)	Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. The possible values for this attribute are defined by the ST_CompoundLine simple type (§20.1.10.15).
w (Line Width)	Specifies the width to be used for the underline stroke. If this attribute is omitted, then a value of 0 is assumed. The possible values for this attribute are defined by the ST_LineWidth simple type (§20.1.10.35).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineProperties) is located in §A.4.1. *end note*]

21.1.3.10 lnTlToBr (Top-Left to Bottom-Right Border Line Properties)

This element defines the line properties associated with the diagonal line from the top left corner of the cell to the bottom right corner.

[*Example*: Consider the following example of a lnTlToBr within DrawingML:


```
<a:lnTlToBr w="38100" cap="flat" cmpd="sng" algn="ctr">
  <a:solidFill>
    <a:schemeClr val="accent2"/>
  </a:solidFill>
  <a:prstDash val="solid"/>
  <a:round/>
  <a:headEnd type="none" w="med" len="med"/>
  <a:tailEnd type="none" w="med" len="med"/>
</a:lnTlToBr>
```

In this example, one can see that the border line style defined with certain properties, such as a flat end line cap, a given width, head and tail end, color, etc. *end example*]

Attributes	Description
algn (Stroke Alignment)	Specifies the alignment to be used for the underline stroke. The possible values for this attribute are defined by the ST_PenAlignment simple type (§20.1.10.39).
cap (Line Ending Cap Type)	Specifies the ending caps that should be used for this line. [*Note*: Examples of cap types are rounded, flat, etc. *end note*] If this attribute is omitted, than a value of square is assumed. The possible values for this attribute are defined by the ST_LineCap simple type (§20.1.10.31).
cmpd (Compound Line Type)	Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. The possible values for this attribute are defined by the ST_CompoundLine simple type (§20.1.10.15).
w (Line Width)	Specifies the width to be used for the underline stroke. If this attribute is omitted, then a value of 0 is assumed. The possible values for this attribute are defined by the ST_LineWidth simple type (§20.1.10.35).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineProperties) is located in §A.4.1. *end note*]

21.1.3.11 tableStyle (Table Style)

This element specifies a particular table style. Fourteen elements make up the styling information of a given table style. These fourteen elements work together to provide visual formatting options for on/off states of the following toggles:

- First row on/off - Associated element: firstRow

- Last row on/off - Associated element: lastRow
- First column on/off - Associated element: firstCol
- Last column on/off - Associated element: lastCol
- Row banding on/off - Associated elements: band1H, band2H
- Column banding on/off - Associated elements: band1V, band2V

The formatting associated with the wholeTbl element defines the table formatting when all options are off. When an option is turned on, the formatting for that particular option is applied to the table. The four cell specific formatting options are enabled when overlapping options are toggled on. For example, when the first row, and first column formatting options are enabled, any formatting within the northwest cell is also be applied since that is the overlapping table cell when both first column and first row formatting options are on.

[*Example*: Consider the following partial example of a tblStyle within DrawingML:

```
<a:tblStyle styleId="{5940675A-B579-460E-94D1-54222C63F5DA}"
  styleName="No Style, Table Grid">
  <a:wholeTbl>
    <a:tcTxStyle>
      <a:fontRef idx="minor">
        <a:scrgbClr r="0" g="0" b="0"/>
      </a:fontRef>
      <a:schemeClr val="tx1"/>
    </a:tcTxStyle>
    <a:tcStyle>
      <a:tcBdr>
        <a:left>
          <a:ln w="12700" cmpd="sng">
            <a:solidFill>
              <a:schemeClr val="tx1"/>
            </a:solidFill>
          </a:ln>
        </a:left>
```

…right, top, bottom, insideH, insideV border information is defined just as the 'left' tag…

```
      </a:tcBdr>
      <a:fill>
        <a:noFill/>
      </a:fill>
    </a:tcStyle>
  </a:wholeTbl>
```



```
<a:band1H>
  <a:tcStyle>
    <a:tcBdr/>
  </a:tcStyle>
</a:band1H>
```

…band2H, band1V, band2V, firstCol, firstRow, lastCol, lastRow, neCell, nwCell, seCell, swCell tags are all defined just as the 'band1H' tag

```
</a:tblStyle>
```

In this example, one can get an idea for the definition of a table style in its entirety. The above defined table style creates a style with only 1pt line formatting applied to all of the cells in a table. Notice that the on/off toggle formatting (band1H, band2H, firstCol, etc) do not define any formatting and therefore have no effect to the table when toggled. *end example*]

Attributes	Description
styleId (Style ID)	Specifies a GUID identifying the table style in a unique manner. The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).
styleName (Name)	Specifies the name of the table style which can show up in the user interface identifying the style to a user. 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_TableStyle) is located in §A.4.1. *end note*]

21.1.3.12 tableStyleId (Table Style ID)

This element defines the table style which is currently applied to the table by referencing the styleId attribute of the tableStyle element.

[*Example*: Consider the following example of a tableStyleId within DrawingML:

```
<a:tblPr firstRow="1" bandRow="1">
  <a:tableStyleId>{5940675A-B579-460E-94D1-54222C63F5DA}</a:tableStyleId>
</a:tblPr>
```

In this example, we see a reference to a table style being specified in the tableStyleId element. *end example*]

The possible values for this element are defined by the ST_Guid simple type (§22.9.2.4).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Guid) is located in §A.6.9. *end note*]

21.1.3.13 tbl (Table)

This element is the root element for a table. Within this element is contained everything that one would need to define a table within DrawingML.

[*Example*: Consider the following example of a tbl within DrawingML:

```
<a:tbl>
  <a:tblPr firstRow="1" bandRow="1">
  …
  <a:tblPr>
  <a:tblGrid>
  …
  </a:tblGrid>
  <a:tr h="419100">
  …
  </a:tr>
</a:tbl>
```

In this example, we see can see the definition of a table within DrawingML. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Table) is located in §A.4.1. *end note*]

21.1.3.14 tblGrid (Table Grid)

This element defines a list of table column (§21.1.3.2) elements. There should be a table column (§21.1.3.2) element for every column held within the table.

[*Example*: Consider the following example of a tblGrid within DrawingML:

```
<a:tblGrid>
  <a:gridCol w="1117600"/>
  <a:gridCol w="1117600"/>
  <a:gridCol w="1117600"/>
</a:tblGrid>
```

In this example, we have a tblGrid defined that holds three columns, therefore the table has three columns. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableGrid) is located in §A.4.1. *end note*]

21.1.3.15 tblPr (Table Properties)

This element defines the properties of a table on the whole. Within this element are many visual modifications that can be applied to the table.

[*Example*: Consider the following example of a tblPr within DrawingML:

```
<a:tblPr firstRow="1" bandRow="1">
  <a:tableStyleId>{5940675A-B579-460E-94D1-54222C63F5DA}</a:tableStyleId>
</a:tblPr>
```

In this example, we see that there is a link to a table style id (§21.1.3.12) which is defined elsewhere and that the first column formatting and banded row formatting has been enabled. The table style defines the formatting applied with the two formatting options enabled. *end example*]

Attributes	Description

bandCol (Banded Columns)	Enables or disables the banded column formatting for a table style. A value of1 or `true` enables the banded column formatting defined in the table style. The attribute defaults to off if it is not specified.
	
	[*Example*: Consider the following run:
	
	`<a:tblPr bandCol="1">`
	`<a:tableStyleId>{5940675A-B579-460E-94D1-`
	`54222C63F5DA}</a:tableStyleId>`
	`</a:tblPr>`
	
	In this example, we can see the banded column formatting is enabled for the table. When applied, the linked table style defines the formatting for banded columns. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

bandRow (Banded Rows)	Enables or disables the banded row formatting for a table style. A value of 1 or `true` enables the banded row formatting defined in the table style. The attribute defaults to `false` if it is not specified.
	
	[*Example*: Consider the following run:
	
	`<a:tblPr bandRow="1">`
	`<a:tableStyleId>{5940675A-B579-460E-94D1-`
	`54222C63F5DA}</a:tableStyleId>`
	`</a:tblPr>`
	
	In this example, we can see the banded row formatting is enabled for the table. When applied, the linked table style defines the formatting for banded rows. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

firstCol (First	Enables or disables the first column formatting for a table style. A value of 1 or `true`

Attributes	Description

Column)	enables the first column formatting defined in the table style. The attribute defaults to `false` if it is not specified.
	
	[*Example*: Consider the following run:
	
	`<a:tblPr firstCol="1">`
	`<a:tableStyleId>{5940675A-B579-460E-94D1-`
	`54222C63F5DA}</a:tableStyleId>`
	`</a:tblPr>`
	
	In this example, we can see the first column formatting is enabled for the table. When applied, the linked table style defines the formatting for the first column. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

firstRow (First Row)	Enables or disables the first row formatting for a table style. A value of `1` or `true` enables the first row formatting defined in the table style. The attribute defaults to `false` if it is not specified.
	
	[*Example*: Consider the following run:
	
	`<a:tblPr firstRow="1">`
	`<a:tableStyleId>{5940675A-B579-460E-94D1-`
	`54222C63F5DA}</a:tableStyleId>`
	`</a:tblPr>`
	
	In this example, we can see the first row formatting is enabled for the table. When applied, the linked table style defines the formatting for the first row. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lastCol (Last Column)	Enables or disables the last column formatting for a table style. A value of `1` or `true` enables the last column formatting defined in the table style. The attribute defaults to `false` if it is not specified.
	
	[*Example*: Consider the following run:
	
	`<a:tblPr lastCol="1">`
	`<a:tableStyleId>{5940675A-B579-460E-94D1-`
	`54222C63F5DA}</a:tableStyleId>`
	`</a:tblPr>`
	
	In this example, we can see the last column formatting is enabled for the table. When applied, the linked table style defines the formatting for the last column. *end example*]

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

lastRow (Last Row)	Enables or disables the last row formatting for a table style. A value of 1 or true enables the last row formatting defined in the table style. The attribute defaults to false if it is not specified.
	
	[*Example*: Consider the following run:
	
	`<a:tblPr lastRow="1">`
	`  <a:tableStyleId>{5940675A-B579-460E-94D1-`
	`  54222C63F5DA}</a:tableStyleId>`
	`</a:tblPr>`
	
	In this example, we can see the last row formatting is enabled for the table. When applied, the linked table style defines the formatting for the last row. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

rtl (Right-to-Left)	Defines enables the right-to-left settings of a table. If the value of rtl is 1 or true , then the table is laid out from the right-to-left rather than the default left-to-right.
	
	[*Example*: Consider the following run:
	
	`<a:tblPr rtl="1">`
	`  <a:tableStyleId>{5940675A-B579-460E-94D1-`
	`  54222C63F5DA}</a:tableStyleId>`
	`</a:tblPr>`
	
	In this example, we can see that the table is to be created in a right-to-left direction. *end example*]
	
	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_TableProperties) is located in §A.4.1. *end note*]

21.1.3.16 tc (Table Cell)

This element defines a cell within the table. The table cell holds a text body that actually contains the data held within the cell along with the properties of the table cell which hold formatting options associated with the cell.

[*Example*: Consider the following example of a tc within DrawingML:

```
<a:tc>
  <a:txBody>
    <a:bodyPr/>
    <a:lstStyle/>
    <a:p>
      <a:pPr marL="0" algn="ctr" rtl="0"/>
      <a:r>
        <a:rPr lang="en-US" dirty="0" smtClean="0"/>
        <a:t>data</a:t>
      </a:r>
      <a:endParaRPr lang="en-US" dirty="0"/>
    </a:p>
  </a:txBody>
  <a:tcPr/>
</a:tc>
```

In this example, we see a single cell in a table being defined with the default cell properties and a text body which contains the word "data". The text "data" is the only text in the cell. *end example*]

Attributes	Description

gridSpan (Grid Span)	Specifies the number of columns that a merged cell spans. This is used in combination with the hMerge attribute on other cells in order to specify the beginning cell of a horizontal merge.
	
	[*Example*: Consider the following example:
	
	`<a:tc gridSpan="3">`
	…
	`/a:tc>`
	`<a:tc hMerge="1">`
	…
	`/a:tc>`
	`<a:tc hMerge="1">`
	…
	`/a:tc>`
	
	In this example, we can define what looks like a single cell in the table as a group of three cells merged together. The merged cell spans three columns of the table. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema int datatype.

hMerge (Horizontal Merge)	When this attribute is set to `1` or `true` , then this table cell is to be merged with the previous horizontal table cell when the table is created. [*Example*: Consider the following example:

Attributes	Description
	<a:tc hMerge="1"> … </a:tc> In this example, we see the hMerge attribute set to on which signifies that this cell is to be merged with the previous horizontal cell in the table. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Table Cell Identifier)	Specifies a unique identifier for the current table cell. This identifier shall be unique within the table, and is used to identify this table cell as a header cell for other cells within the table, using the headers child element. If this element is omitted, this table cell has no unique identifier. [*Example*: Consider a table cell defined as follows: <a:tc id="januarytwentynine"> … </a:tc> The value in the id specifies a unique identifer oif januarytwentynine. Other cells in the table are then able to reference this cell as a row or column header by referencing this ID. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
rowSpan (Row Span)	Specifies the number of rows that a merged cell spans. This is used in combination with the vMerge attribute on other cells in order to specify the beginning cell of a horizontal merge. [*Example*: Consider the following example: <a:tc rowSpan="3"> … /a:tc> <a:tc vMerge="1"> … /a:tc> <a:tc vMerge="1"> … /a:tc> In this example, we can define what looks like a single cell in the table as a group of three

Attributes	Description
	cells merged together. The merged cell spans three rows of the table. *end example*] The possible values for this attribute are defined by the W3C XML Schema int datatype.

vMerge (Vertical Merge)	When this attribute is set to 1 or true , then this table cell is to be merged with the previous vertical table cell when the table is created.
	
	[*Example*: Consider the following example:
	
	`<a:tc vMerge="1">`
	`…`
	`/a:tc>`
	
	In this example, we see the vMerge attribute set to on which signifies that this cell is to be merged with the previous vertical cell in the table. *end example*]
	
	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_TableCell) is located in §A.4.1. *end note*]

21.1.3.17 tcPr (Table Cell Properties)

This element defines the formatting properties associated with a cell. The formatting options which are available to be adjusted range from the line types used for the borders to the cell fill to the margins associated with the layout of the text in the cell.

[*Example*: Consider the following example of a tcPr within DrawingML:

```
<a:tcPr marL="45720" marR="45720">
  <a:lnL w="38100" cap="flat" cmpd="sng" algn="ctr">
    <a:solidFill>
      <a:schemeClr val="accent2"/>
    </a:solidFill>
    <a:prstDash val="solid"/>
    <a:round/>
    <a:headEnd type="none" w="med" len="med"/>
    <a:tailEnd type="none" w="med" len="med"/>
  </a:lnL>
</a:tcPr>
```


In this example, we have a solid line defined as the left border of the cell along with left and right margin adjustments being made from the default margins. *end example*]

Attributes	Description
anchor (Anchor)	Defines the alignment of the text vertically within the cell. [*Example*: Consider the following example: `<a:tcPr marL="45720" anchor="ctr"/>` In this example, the text in the cell is anchored to the center of the cell vertically. *end example*] The possible values for this attribute are defined by the ST_TextAnchoringType simple type (§20.1.10.59).
anchorCtr (Anchor Center)	When this attribute is `1` or `true` , it modifies the anchor attribute. This attribute center-aligns the text box itself which allows for text to be left aligned along the center of the cell for example. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

horzOverflow (Horizontal Overflow)	Specifies the clipping behavior of the cell. The two options here allow for the text to be clipped and out of view when outside of the bounds of the cell, or for the text to remain visible and overflow outside of the cell.
	
	[*Example*: Consider the following example:
	
	`<a:tcPr horzOverflow="overflow">`
	`…`
	`</a:tcPr>`
	
	In this example, the text in the cell freely overflows outside of the cell boundaries and always remains visible. *end example*]
	
	The possible values for this attribute are defined by the ST_TextHorzOverflowType simple type (§20.1.10.68).

marB (Bottom Margin)	Specifies the bottom margin of the cell. The value specified in this attribute is the distance to offset from the bottom of the cell.
	
	[*Example*: Consider the following example:
	
	`<a:tcPr marB="45720" anchor="ctr">`
	`…`
	`</a:tcPr>`
	
	In this example, we have specified a value for the margin on the bottom of the cell. *end*

Attributes	Description
	example] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).

marL (Left Margin)	This attribute specifies the left margin of the cell. The value specified in this attribute is the distance to offset from the left of the cell in EMU's.
	
	[*Example*: Consider the following example:
	
	`<a:tcPr marL="45720" anchor="ctr">`
	`…`
	`</a:tcPr>`
	
	In this example, we have specified a value for the margin on the left of the cell. *end example*]
	
	The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).

marR (Right Margin)	This attribute specifies the right margin of the cell. The value specified in this attribute is the distance to offset from the right of the cell in EMU's.
	
	[*Example*: Consider the following example:
	
	`<a:tcPr marR="45720" anchor="ctr">`
	`…`
	`</a:tcPr>`
	
	In this example, we have specified a value for the margin on the right of the cell. *end example*]
	
	The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).

marT (Top Margin)	This attribute specifies the top margin of the cell. The value specified in this attribute is the distance to offset from the top of the cell in EMU's.
	
	[*Example*: Consider the following example:
	
	`<a:tcPr marT="45720" anchor="ctr">`
	`…`
	`</a:tcPr>`
	
	In this example, we have specified a value for the margin on the top of the cell. *end example*]

Attributes	Description
	The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).

vert (Text Direction)	Defines the text direction within the cell.
	
	[*Example*: Consider the following example:
	`<a:tcPr vert="vert270">`
	…
	`</a:tcPr>`
	
	In this example, we have rotated the layout of the text 270 degrees so that it starts at the bottom of the cell and goes upward toward the top of the cell. *end example*]
	
	The possible values for this attribute are defined by the ST_TextVerticalType simple type (§20.1.10.82).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableCellProperties) is located in §A.4.1. *end note*]

21.1.3.18 tr (Table Row)

This element defines a row in a table. A row as defined in a table is simply a listing of table cells (§21.1.3.16). There is a table row element defined for every row in the table.

[*Example*: Consider the following example of a tr within DrawingML:

```
<a:tr h="774700">
  <a:tc>
    <a:txBody>
      <a:bodyPr/>
      <a:lstStyle/>
      <a:p>
        <a:endParaRPr lang="en-US" dirty="0"/>
      </a:p>
    </a:txBody>
    <a:tcPr/>
  </a:tc>
…
</a:tr>
```

In this example, we see a table row defined with an example table cell (§21.1.3.16) defined within it. The height of the row has been specified and in real use, there is a table cell defined in this row for each grid column (§21.1.3.2) defined in the table. *end example*]

Attributes	Description
h (Height)	Defines the height of the row in the table. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableRow) is located in §A.4.1. *end note*]

21.2 DrawingML - Charts

The chart namespace in DrawingML is for representing visualizations of numeric data with column charts, pie charts, scatter charts, or other types of charts.

21.2.1 Table of Contents

This subclause is informative.

End of informative text.

21.2.2 Elements

In DrawingML, charts define a visualization of numeric data. The definition includes where the data shall come from, a cache of the data, and how the data shall be represented graphically. Other DrawingML elements are reused to define aspects of the formatting of the visualization.

See the informative material in Annex L for a description and overview of the basic chart types and chart components.

21.2.2.1 applyToEnd (Apply to End)

This element specifies the picture shall be applied to the end of the point or series.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.2 applyToFront (Apply To Front)

This element specifies the picture shall be applied to the front of the point or series.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.3 applyToSides (Apply To Sides)

This element specifies the picture shall be applied to the sides of the point or series.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.4 area3DChart (3D Area Charts)

This element specifies the 3-D area series on this chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Area3DChart) is located in §A.5.1. *end note*]

21.2.2.5 areaChart (Area Charts)

This element specifies the 2-D area series on this chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AreaChart) is located in §A.5.1. *end note*]

21.2.2.6 auto (Automatic Category Axis)

This element specifies that this axis is a date or text axis based on the data that is used for the axis labels, not a specific choice.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.7 autoTitleDeleted (Auto Title Is Deleted)

This element specifies the title shall not be shown for this chart.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied.

Attributes	Description
	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_Boolean) is located in §A.5.1. *end note*]

21.2.2.8 autoUpdate (Update Automatically)

This element specifies the external data is updated automatically when the document containing the chart is opened.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.9 axId (Axis ID)

When specified as a child element of valAx, dateAx, catAx, or serAx, this element specifies the identifier for the axis. When specified as a child element of a chart, this element specifies the identifier of an axis that defines the coordinate space of the chart.

Attributes	Description
val (Integer Value)	Specifies that the contents of this attribute contain an integer number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnsignedInt) is located in §A.5.1. *end note*]

21.2.2.10 axPos (Axis Position)

This element specifies the position of the axis on the chart.

Attributes	Description
val (Axis Position Value)	Specifies the position of the axis on the chart. The possible values for this attribute are defined by the ST_AxPos simple type (§21.2.3.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AxPos) is located in §A.5.1. *end note*]

21.2.2.11 backWall (Back Wall)

This element specifies the back wall of the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Surface) is located in §A.5.1. *end note*]

21.2.2.12 backward (Backward)

This element specifies the number of categories (or units on a scatter chart) that the trend line extends before the data for the series that is being trended. On scatter and non-scatter charts, the value shall be any non-negative value.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.13 bandFmt (Band Format)

This element specifies the formatting band of a surface chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BandFmt) is located in §A.5.1. *end note*]

21.2.2.14 bandFmts (Band Formats)

This element contains a collection of formatting bands for a surface chart indexed from low to high.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BandFmts) is located in §A.5.1. *end note*]

21.2.2.15 bar3DChart (3D Bar Charts)

This element contains the 3-D bar or column series on this chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Bar3DChart) is located in §A.5.1. *end note*]

21.2.2.16 barChart (Bar Charts)

This element contains the 2-D bar or column series on this chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BarChart) is located in §A.5.1. *end note*]

21.2.2.17 barDir (Bar Direction)

This element specifies whether the series form a bar (horizontal) chart or a column (vertical) chart

Attributes	Description
val (Bar Direction Value)	Specifies the direction of the series. The possible values for this attribute are defined by the ST_BarDir simple type (§21.2.3.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BarDir) is located in §A.5.1. *end note*]

21.2.2.18 baseTimeUnit (Base Time Unit)

This element specifies the smallest time unit that is represented on the date axis.

Attributes	Description
val (Time Unit Value)	Specifies the time unit for the tick marks. The possible values for this attribute are defined by the ST_TimeUnit simple type (§21.2.3.49).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TimeUnit) is located in §A.5.1. *end note*]

21.2.2.19 bubble3D (3D Bubble)

This element specifies that the bubbles have a 3-D effect applied to them.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.20 bubbleChart (Bubble Charts)

This element contains the bubble series on this chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BubbleChart) is located in §A.5.1. *end note*]

21.2.2.21 bubbleScale (Bubble Scale)

This element specifies the scale factor for the bubble chart. This element can be a percentage value from 0 to 300, corresponding to a percentage of the default size.

Attributes	Description
val (Bubble Scale Value)	Specifies how to scale bubbles on a bubble chart. The possible values for this attribute are defined by the ST_BubbleScale simple type (§21.2.3.5).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BubbleScale) is located in §A.5.1. *end note*]

21.2.2.22 bubbleSize (Bubble Size)

This element specifies the data for the sizes of the bubbles on the bubble chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumDataSource) is located in §A.5.1. *end note*]

21.2.2.23 builtInUnit (Built in Display Unit Value)

This element specifies the display unit is one of the built in values.

Attributes	Description
val (Built In Unit Value)	Specifies the display unit scaling applied to the axis. The possible values for this attribute are defined by the ST_BuiltInUnit simple type (§21.2.3.6).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BuiltInUnit) is located in §A.5.1. *end note*]

21.2.2.24 cat (Category Axis Data)

This element specifies the data used for the category axis.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AxDataSource) is located in §A.5.1. *end note*]

21.2.2.25 catAx (Category Axis Data)

This element specifies the category axis of the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CatAx) is located in §A.5.1. *end note*]

21.2.2.26 chart (Reference to Chart Part)

This element specifies the chart.

Attributes	Description
id (Relationship Reference) Namespace: http://purl.oclc.org/ooxml/officeDocument/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 (§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*]

21.2.2.27 chart (Chart)

This element specifies the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Chart) is located in §A.5.1. *end note*]

21.2.2.28 chartObject (Chart Object)

This element specifies that the chart cannot be edited by the user

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.29 chartSpace (Chart Space)

This element specifies overall settings for a single chart, and is the root node for the chart part.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChartSpace) is located in §A.5.1. *end note*]

21.2.2.30 clrMapOvr (Color Map Override)

This element represents color mapping information. It is used to override the applications color mapping if the user has selected keep source formatting after a copy-paste.

Attributes	Description
accent1 (Accent 1) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.14).
accent2 (Accent 2)	Specifies a color defined which is associated as the accent 2 color.

Attributes	Description
Namespace: http://purl.oclc.org/ooxml/drawingml/main	The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent3 (Accent 3) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.14).
accent4 (Accent 4) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.14).
accent5 (Accent 5) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.14).
accent6 (Accent 6) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.14).
bg1 (Background 1) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.14).
bg2 (Background 2) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.14).
folHlink (Followed Hyperlink) Namespace:	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 (§20.1.10.14).

Attributes	Description
http://purl.oclc.org/ooxml/drawingml/main	
hlink (Hyperlink) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.14).
tx1 (Text 1) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.14).
tx2 (Text 2) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorMapping) is located in §A.4.1. *end note*]

21.2.2.31 crossAx (Crossing Axis ID)

This element specifies the ID of axis that this axis crosses. For instance, a category axis might cross a value axis, and the category axis's crossAx would contain the ID of the value axis.

Attributes	Description
val (Integer Value)	Specifies that the contents of this attribute contain an integer number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnsignedInt) is located in §A.5.1. *end note*]

21.2.2.32 crossBetween (Cross Between)

This element specifies whether the value axis crosses the category axis between categories.

If not specified, then the application should choose an appropriate behavior.

Attributes	Description
val (Cross Between Value)	Specifies whether the value axis crosses the category axis between categories or on categories. The possible values for this attribute are defined by the ST_CrossBetween simple type (§21.2.3.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_CrossBetween) is located in §A.5.1. *end note*]

21.2.2.33 crosses (Crosses)

This element specifies how this axis crosses the perpendicular axis.

Attributes	Description
val (Crosses Value)	Specifies where the axis crosses its perpendicular axis. The possible values for this attribute are defined by the ST_Crosses simple type (§21.2.3.8).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Crosses) is located in §A.5.1. *end note*]

21.2.2.34 crossesAt (Crossing Value)

This element specifies where on the axis the perpendicular axis crosses. The units are dependent on the type of axis.

When specified as a child element of valAx, the value is a decimal number on the value axis. When specified as a child element of dateAx, the date is defined as a integer number of days relative to the base date of the current date system. When specified as a child element of catAx, the value is an integer category number, starting with 1 as the first category.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element.

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

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.35 custSplit (Custom Split)

This element contains the custom split information for a pie-of-pie or bar-of-pie chart with a custom split.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustSplit) is located in §A.5.1. *end note*]

21.2.2.36 custUnit (Custom Display Unit)

This element specifies a custom value for the display unit.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.37 data (Data Cannot Be Changed)

This element specifies that the user cannot change the choice of data used for the chart

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.38 date1904 (1904 Date System)

This element specifies that the chart uses the 1904 date system. If the 1904 date system is used, then all dates and times shall be specified as a decimal number of days since Dec. 31, 1903. If the 1904 date system is not used, then all dates and times shall be specified as a decimal number of days since Dec. 31, 1899.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.39 dateAx (Date Axis)

This element specifies a date axis for the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DateAx) is located in §A.5.1. *end note*]

21.2.2.40 delete (Delete)

This element specifies that the chart element specified by its containing element shall be deleted from the chart.

This should be set to true if the application adds these elements by default even after the user has specified that they should be removed from the chart.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted.

Attributes	Description
	A value of `0` or `false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.41 depthPercent (Depth Percent)

This element specifies the depth of a 3-D chart as a percentage of the chart width (between 20 and 2000 percent).

Attributes	Description
val (Depth Percent Value)	Specifies a percentage value for the property defined by the parent XML element. The possible values for this attribute are defined by the ST_DepthPercent simple type (§21.2.3.9).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DepthPercent) is located in §A.5.1. *end note*]

21.2.2.42 dispBlanksAs (Display Blanks As)

This element specifies how blank cells shall be plotted on a chart .

Attributes	Description
val (Display Blanks As Value)	Specifies how blank cells are plotted on the chart. The possible values for this attribute are defined by the ST_DispBlanksAs simple type (§21.2.3.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DispBlanksAs) is located in §A.5.1. *end note*]

21.2.2.43 dispEq (Display Equation)

This element specifies that the equation for the trendline is displayed on the chart (in the same label as the R-squared value).

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element.

Attributes	Description
	A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.44 dispRSqr (Display R Squared Value)

This element specifies that the R-squared value of the trendline is displayed on the chart (in the same label as the equation).

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.45 dispUnits (Display Units)

This element specifies the scaling value of the display units for the value axis.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DispUnits) is located in §A.5.1. *end note*]

21.2.2.46 dispUnitsLbl (Display Units Label)

This element specifies the display unit label for the value axis in the specified chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DispUnitsLbl) is located in §A.5.1. *end note*]

21.2.2.47 dLbl (Data Label)

This element specifies a data label.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DLbl) is located in §A.5.1. *end note*]

21.2.2.48 dLblPos (Data Label Position)

This element specifies the position of the data label.

Attributes	Description
val (Data Label Position Value)	Specifies how the data label is positioned on the chart. The possible values for this attribute are defined by the ST_DLblPos simple type (§21.2.3.11).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DLblPos) is located in §A.5.1. *end note*]

21.2.2.49 dLbls (Data Labels)

This element serves as a root element that specifies the settings for the data labels for an entire series or the entire chart. It contains child elements that specify the specific formatting and positioning settings.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DLbls) is located in §A.5.1. *end note*]

21.2.2.50 doughnutChart (Doughnut Charts)

This element contains the doughnut chart series.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DoughnutChart) is located in §A.5.1. *end note*]

21.2.2.51 downBars (Down Bars)

This element specifies the down bars.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UpDownBar) is located in §A.5.1. *end note*]

21.2.2.52 dPt (Data Point)

This element specifies a single data point.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DPt) is located in §A.5.1. *end note*]

21.2.2.53 dropLines (Drop Lines)

This element specifies drop lines.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChartLines) is located in §A.5.1. *end note*]

21.2.2.54 dTable (Data Table)

This element specifies a data table.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DTable) is located in §A.5.1. *end note*]

21.2.2.55 errBars (Error Bars)

This element specifies error bars. The errValType element controls whether the minus, plus, or val elements are used.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ErrBars) is located in §A.5.1. *end note*]

21.2.2.56 errBarType (Error Bar Type)

This element specifies the style of the error bars - positive, negative, or both.

Attributes	Description
val (Error Bar Type Value)	Specifies the style of error bars. The possible values for this attribute are defined by the ST_ErrBarType simple type (§21.2.3.12).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ErrBarType) is located in §A.5.1. *end note*]

21.2.2.57 errDir (Error Bar Direction)

This element specifies the direction of the error bars.

Attributes	Description
val (Error Bar Direction Value)	Specifies the direction of the error bars. The possible values for this attribute are defined by the ST_ErrDir simple type (§21.2.3.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ErrDir) is located in §A.5.1. *end note*]

21.2.2.58 errValType (Error Bar Value Type)

This element specifies the type of values used to determine the length of the error bars.

Attributes	Description
val (Error Bar Type Value)	Specifies the type of values of the error bars. The possible values for this attribute are defined by the ST_ErrValType simple type (§21.2.3.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ErrValType) is located in §A.5.1. *end note*]

21.2.2.59 evenFooter (Even Footer)

This element specifies the footer to use on even numbered pages.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

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. 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 (ST_Xstring) is located in §A.6.9. *end note*]

21.2.2.60 evenHeader (Even Header)

This element specifies the header to use on even numbered pages.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

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. 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 (ST_Xstring) is located in §A.6.9. *end note*]

21.2.2.61 explosion (Explosion)

This element specifies the amount the data point shall be moved from the center of the pie.

Attributes	Description
val (Integer Value)	Specifies that the contents of this attribute contain an integer number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnsignedInt) is located in §A.5.1. *end note*]

21.2.2.62 ext (Extension)

This element specifies an extension that is used for future extensions to the current version of DrawingML. This allows for the specifying of currently unknown elements in the future that are used for later versions of generating applications.

Attributes	Description
uri (Uniform Resource Identifier)	Specifies the URI, or uniform resource identifier that represents the data stored under this tag. The URI is used to identify the correct 'server' that can process the contents of this tag. The possible values for this attribute are defined by the W3C XML Schema token datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Extension) is located in §A.5.1. *end note*]

21.2.2.63 externalData (External Data Relationship)

This element specifies the relationship to the data for this chart.

The data can be linked, pointing to a spreadsheet in another file. Or, the data can be embedded, contained in a separate part within the same xml package containing the chart. In this case, it shall be stored as an embedded Spreadsheet object in Office Open XML format.

This is not used by a spreadsheet application as the spreadsheet application can maintain its own reference to the data in the spreadsheet via the formula `<f>` element.

Attributes	Description
id (Relationship Reference) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID for the relationship for this chart. The relationship explicitly targeted by this attribute shall either be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/package, or http://schemas.openxmlformats.org/officeDocument/2006/relationships/oleObject. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ExternalData) is located in §A.5.1. *end note*]

21.2.2.64 extLst (Chart Extensibility)

This element contains tags used for future extensibility of the file format.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ExtensionList) is located in §A.5.1. *end note*]

21.2.2.65 f (Formula)

This element specifies a reference to source of the data contained in this chart. This shall be used by the spreadsheet application only. A presentation, or word processing application should use the `externalData` element.

This reference is in the form of a book, sheet, and cell reference or a book, optional sheet, and defined name reference. This reference does not include the equals sign. [*Example*:


```
<c:cat>
  <c:strRef>
    <c:f>Sheet1!$A$1:$C$1</c:f>
    <c:strCache>
…
    </c:strCache>
  </c:strRef>
</c:cat>
```

The above example shows a formula reference used for the string cahce. In this case the series names, which are referenced by the formula element, are in cells A1, B1, and C1.

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

21.2.2.66 firstFooter (First Footer)

This element specifies the footer to use on the first page.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

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. 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 (ST_Xstring) is located in §A.6.9. *end note*]

21.2.2.67 firstHeader (First Header)

This element specifies the header to use on the first page.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

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. 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 (ST_Xstring) is located in §A.6.9. *end note*]

21.2.2.68 firstSliceAng (First Slice Angle)

This element specifies the angle of the first pie or doughnut chart slice, in degrees (clockwise from up).

Attributes	Description
val (First Slice Angle Value)	Specifies the angle of the first slice. The possible values for this attribute are defined by the ST_FirstSliceAng simple type (§21.2.3.15).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FirstSliceAng) is located in §A.5.1. *end note*]

21.2.2.69 floor (Floor)

This element specifies the floor of a 3D chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Surface) is located in §A.5.1. *end note*]

21.2.2.70 fmtId (Format ID)

This element represents a pivot format ID. It serves as a link back to the correct pivotTable which in turn specifies a link that then defines which set of chart format rules apply to this chart.

This ID shall match the chartFormat element, `chart` attribute, described in §18.10.1.12 of the SpreadsheetML reference material. The chartFormat element also contains a `format` attribute which is used to index into the pivotFmts collection (§21.2.2.143).

Attributes	Description
val (Integer Value)	Specifies that the contents of this attribute contain an integer number.

Attributes	Description
	The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnsignedInt) is located in §A.5.1. *end note*]

21.2.2.71 formatCode (Format Code)

This element specifies a string representing the format code to apply. For more information see the SpreadsheetML numFmt element's (§18.8.30) formatCode attribute.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

21.2.2.72 formatting (Formatting)

This element specifies that a user cannot change formatting on chart elements.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.73 forward (Forward)

This element specifies the number of categories (or units on a scatter chart) that the trendline extends after the data for the series that is being trended. On scatter and non-scatter charts, the value shall be any non-negative value.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.74 gapDepth (Gap Depth)

This element specifies the space between bar or column clusters, as a percentage of the bar or column width.

Attributes	Description
val (Gap Size Value)	Specifies that the contents of this attribute contain a gap amount between 0% and 500%. The possible values for this attribute are defined by the ST_GapAmount simple type (§21.2.3.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GapAmount) is located in §A.5.1. *end note*]

21.2.2.75 gapWidth (Gap Width)

This element specifies the space between bar or column clusters, as a percentage of the bar or column width.

Attributes	Description
val (Gap Size Value)	Specifies that the contents of this attribute contain a gap amount between 0% and 500%. The possible values for this attribute are defined by the ST_GapAmount simple type (§21.2.3.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GapAmount) is located in §A.5.1. *end note*]

21.2.2.76 grouping (Grouping)

This element specifies the kind of grouping for a column, line, or area chart.

Attributes	Description
val (Grouping	Specifies the grouping value.

Attributes	Description
Value)	The possible values for this attribute are defined by the ST_Grouping simple type (§21.2.3.17).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Grouping) is located in §A.5.1. *end note*]

21.2.2.77 grouping (Bar Grouping)

This element specifies the kind of grouping for a bar chart.

Attributes	Description
val (Bar Grouping Value)	Specifies the bar grouping value. The possible values for this attribute are defined by the ST_BarGrouping simple type (§21.2.3.4).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BarGrouping) is located in §A.5.1. *end note*]

21.2.2.78 h (Height)

This element specifies the height (if Height Mode is Factor) or bottom (if Height Mode is edge) of the chart element as a fraction of the height of the chart.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.79 headerFooter (Header and Footer)

This element specifies the headers and footers that shall be used when the chart is printed.

Attributes	Description
alignWithMargins	Specifies the header and footer should align with the left and right margins of the chart.

Attributes	Description
(Align With Margins)	A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
differentFirst (Different First)	Specifies the header and footer are different for the first page. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
differentOddEven (Different Odd Even)	Specifies the header and footer are different on odd-numbered pages and even-numbered pages. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_HeaderFooter) is located in §A.5.1. *end note*]

21.2.2.80 hiLowLines (High Low Lines)

This element specifies the high-low lines for the series.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChartLines) is located in §A.5.1. *end note*]

21.2.2.81 hMode (Height Mode)

This element specifies how to interpret the Height element for this manual layout.

Attributes	Description
val (Layout Mode Value)	Specifies the layout mode for the width. The possible values for this attribute are defined by the ST_LayoutMode simple type (§21.2.3.20).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LayoutMode) is located in §A.5.1. *end note*]

21.2.2.82 holeSize (Hole Size)

This element specifies the size of the hole in a doughnut chart group.

Attributes	Description
val (Hole Size Value)	Specifies that the contents of this attribute contain a hole size between 10% and 90% of the size of the plot area. The possible values for this attribute are defined by the ST_HoleSize simple type (§21.2.3.18).

[*Note*: The W3C XML Schema definition of this element's content model (CT_HoleSize) is located in §A.5.1. *end note*]

21.2.2.83 hPercent (Height Percent)

This element specifies the height of a 3-D chart as a percentage of the chart width.

Attributes	Description
val (Height Percent Value)	Specifies that the contents of this attribute contain a height percent between 5% and 500%. The possible values for this attribute are defined by the ST_HPercent simple type (§21.2.3.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_HPercent) is located in §A.5.1. *end note*]

21.2.2.84 idx (Index)

This element specifies the index of the containing element. This index shall determine which of the parent's children collection this element applies to.

Attributes	Description
val (Integer Value)	Specifies that the contents of this attribute contain an integer number.

Attributes	Description
	The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnsignedInt) is located in §A.5.1. *end note*]

21.2.2.85 intercept (Intercept)

This element specifies the value where the trendline shall cross the y axis. This property shall be supported only when the trendline type is exp, linear, or poly.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.86 invertIfNegative (Invert if Negative)

This element specifies the parent element shall invert its colors if the value is negative.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.87 lang (Editing Language)

This element specifies the primary editing language which was use when this chart was last modified.

Attributes	Description
val (Language Code)	Specifies a language tag as defined by RFC 3066. See simple type for additional information. The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextLanguageID) is located in §A.5.1. *end note*]

21.2.2.88 layout (Layout)

This element specifies how the chart element is placed on the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Layout) is located in §A.5.1. *end note*]

21.2.2.89 layoutTarget (Layout Target)

This element specifies whether to layout the plot area by its inside (not including axis and axis labels) or outside (including axis and axis labels).

Attributes	Description
val (Layout Target Value)	Specifies the layout target value. The possible values for this attribute are defined by the ST_LayoutTarget simple type (§21.2.3.21).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LayoutTarget) is located in §A.5.1. *end note*]

21.2.2.90 lblAlgn (Label Alignment)

This element specifies the text alignment for the tick labels on the axis.

Attributes	Description
val (Label Alignment Value)	Specifies the label alignment. The possible values for this attribute are defined by the ST_LblAlgn simple type (§21.2.3.22).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LblAlgn) is located in §A.5.1. *end note*]

21.2.2.91 lblOffset (Label Offset)

This element specifies the distance of labels from the axis.

Attributes	Description
val (Label Offset Value)	Specifies the distance of labels from the axis. Shall contain a percentage between 0% and 1000%. The possible values for this attribute are defined by the ST_LblOffset simple type (§21.2.3.23).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LblOffset) is located in §A.5.1. *end note*]

21.2.2.92 leaderLines (Leader Lines)

This element specifies the leader lines for data labels.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChartLines) is located in §A.5.1. *end note*]

21.2.2.93 legend (Legend)

This element specifies the legend.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Legend) is located in §A.5.1. *end note*]

21.2.2.94 legendEntry (Legend Entry)

This element specifies a legend entry.

[*Note*: The W3C XML Schema definition of this element's content model (CT_LegendEntry) is located in §A.5.1. *end note*]

21.2.2.95 legendPos (Legend Position)

This element specifies the position of the legend.

Attributes	Description
val (Legend Position Value)	Specifies the position of the legend. The possible values for this attribute are defined by the ST_LegendPos simple type

Attributes	Description
	(§21.2.3.24).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LegendPos) is located in §A.5.1. *end note*]

21.2.2.96 line3DChart (3D Line Charts)

This element contains the 3-D line chart series.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Line3DChart) is located in §A.5.1. *end note*]

21.2.2.97 lineChart (Line Charts)

This element contains the 2-D line chart series.

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineChart) is located in §A.5.1. *end note*]

21.2.2.98 logBase (Logarithmic Base)

This element specifies the logarithmic base for a logarithmic axis.

Attributes	Description
val (Logarithmic Base Value)	Specifies the logarithmic base for a logarithmic axis. Shall contain a floating point value greater than or equal to 2. The possible values for this attribute are defined by the ST_LogBase simple type (§21.2.3.25).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LogBase) is located in §A.5.1. *end note*]

21.2.2.99 lvl (Level)

This element specifies data for a single level of labels for a category axis.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Lvl) is located in §A.5.1. *end note*]

21.2.2.100 majorGridlines (Major Gridlines)

This element specifies major gridlines.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChartLines) is located in §A.5.1. *end note*]

21.2.2.101 majorTickMark (Major Tick Mark)

This element specifies the major tick marks.

Attributes	Description
val (Tick Mark Value)	Specifies the minor tick mark position. The possible values for this attribute are defined by the ST_TickMark simple type (§21.2.3.48).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TickMark) is located in §A.5.1. *end note*]

21.2.2.102 majorTimeUnit (Major Time Unit)

This element specifies the time unit for major tick marks.

Attributes	Description
val (Time Unit Value)	Specifies the time unit for the tick marks. The possible values for this attribute are defined by the ST_TimeUnit simple type (§21.2.3.49).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TimeUnit) is located in §A.5.1. *end note*]

21.2.2.103 majorUnit (Major Unit)

This element specifies the distance between major ticks.

Attributes	Description
val (Major Unit Value)	Specifies the distance between major ticks. Shall contain a positive floating-point number. The possible values for this attribute are defined by the ST_AxisUnit simple type (§21.2.3.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AxisUnit) is located in §A.5.1. *end note*]

21.2.2.104 manualLayout (Manual Layout)

This element specifies the exact position of a chart element.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ManualLayout) is located in §A.5.1. *end note*]

21.2.2.105 marker (Show Marker)

This element is a Boolean that, when `true`, specifies that the marker shall be shown.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.106 marker (Marker)

This element specifies a data marker.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Marker) is located in §A.5.1. *end note*]

21.2.2.107 max (Maximum)

This element specifies the maximum value of the axis.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.108 min (Minimum)

This element specifies the minimum value of the axis.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.109 minorGridlines (Minor Gridlines)

This element specifies the minor gridlines.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChartLines) is located in §A.5.1. *end note*]

21.2.2.110 minorTickMark (Minor Tick Mark)

This element specifies the minor tick marks for the axis.

Attributes	Description
val (Tick Mark Value)	Specifies the minor tick mark position. The possible values for this attribute are defined by the ST_TickMark simple type (§21.2.3.48).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TickMark) is located in §A.5.1. *end note*]

21.2.2.111 minorTimeUnit (Minor Time Unit)

This element specifies the time unit for the minor tick marks.

Attributes	Description
val (Time Unit Value)	Specifies the time unit for the tick marks. The possible values for this attribute are defined by the ST_TimeUnit simple type (§21.2.3.49).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TimeUnit) is located in §A.5.1. *end note*]

21.2.2.112 minorUnit (Minor Unit)

This element specifies the distance between minor tick marks.

Attributes	Description
val (Minor Unit Value)	Specifies the distance between minor tick marks. Shall contain a positive floating-point number. The possible values for this attribute are defined by the ST_AxisUnit simple type (§21.2.3.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AxisUnit) is located in §A.5.1. *end note*]

21.2.2.113 minus (Minus)

This element specifies the error bar value in the negative direction. It shall be used only when the errValType is cust.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumDataSource) is located in §A.5.1. *end note*]

21.2.2.114 multiLvlStrCache (Multi Level String Cache)

This element specifies the last data shown on the chart for a category axis.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MultiLvlStrData) is located in §A.5.1. *end note*]

21.2.2.115 multiLvlStrRef (Multi Level String Reference)

This element specifies a reference to data for the category axis with a cache of the last values used.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MultiLvlStrRef) is located in §A.5.1. *end note*]

21.2.2.116 name (Trendline Name)

This element specifies the name of the trendline.

The possible values for this element are defined by the W3C XML Schema string datatype.

21.2.2.117 name (Pivot Name)

This element specifies the name of the pivot table to get the data for the chart from.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

21.2.2.118 noEndCap (No End Cap)

This element specifies an end cap is not drawn on the error bars.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.119 noMultiLvlLbl (No Multi-level Labels)

This element specifies the labels shall be shown as flat text. If this element is not included or is set to false, then the labels shall be drawn as a hierarchy.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.120 numCache (Number Cache)

This element specifies the last data shown on the chart for a series.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumData) is located in §A.5.1. *end note*]

21.2.2.121 numFmt (Number Format)

This element specifies number formatting for the parent element.

Attributes	Description
formatCode (Number Format Code)	This element specifies a string representing the format code to apply. For more information see the SpreadsheetML numFmt element's (§18.8.30) formatCode attribute. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
sourceLinked (Linked to Source)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_NumFmt) is located in §A.5.1. *end note*]

21.2.2.122 numLit (Number Literal)

This element specifies a set of numbers used for the parent element.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumData) is located in §A.5.1. *end note*]

21.2.2.123 numRef (Number Reference)

This element specifies a reference to numeric data with a cache of the last values used.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumRef) is located in §A.5.1. *end note*]

21.2.2.124 oddFooter (Odd Footer)

This element specifies the footer to use on odd numbered pages.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

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. 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 (ST_Xstring) is located in §A.6.9. *end note*]

21.2.2.125 oddHeader (Odd Header)

This element specifies the header to use on odd numbered pages.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

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. 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 (ST_Xstring) is located in §A.6.9. *end note*]

21.2.2.126 ofPieChart (Pie of Pie or Bar of Pie Charts)

This element contains the pie of pie or bar of pie series on this chart. Only the first series shall be displayed. The splitType element shall determine whether the splitPos and custSplit elements apply.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OfPieChart) is located in §A.5.1. *end note*]

21.2.2.127 ofPieType (Pie of Pie or Bar of Pie Type)

This element specifies whether this chart is pie of pie or bar of pie.

Attributes	Description
val (Pie of Pie or Bar of Pie Type Value)	Specifies the type of pie of pie or bar of pie chart. The possible values for this attribute are defined by the ST_OfPieType simple type (§21.2.3.28).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OfPieType) is located in §A.5.1. *end note*]

21.2.2.128 order (Order)

This element specifies the order of the series in the collection. It is 0 based.

Attributes	Description
val (Integer Value)	Specifies that the contents of this attribute contain an integer number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnsignedInt) is located in §A.5.1. *end note*]

21.2.2.129 order (Polynomial Trendline Order)

This element specifies the order of the polynomial trend line. It is ignored for other trend line types.

Attributes	Description
val (Order Value)	Specifies that the contents of this attribute contain an integer between 2 and 6. The possible values for this attribute are defined by the ST_Order simple type (§21.2.3.29).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Order) is located in §A.5.1. *end note*]

21.2.2.130 orientation (Axis Orientation)

This element specifies the stretching and stacking of the picture on the data point, series, wall, or floor.

Attributes	Description
val (Orientation Value)	Specifies the orientation of the axis. The possible values for this attribute are defined by the ST_Orientation simple type (§21.2.3.30).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Orientation) is located in §A.5.1. *end note*]

21.2.2.131 overlap (Overlap)

This element specifies how much bars and columns shall overlap on 2-D charts.

Attributes	Description
val (Overlap Value)	Specifies how much bars and columns shall overlap on 2-D charts. Shall contain a percentage between -100% and 100%. The possible values for this attribute are defined by the ST_Overlap simple type (§21.2.3.31).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Overlap) is located in §A.5.1. *end note*]

21.2.2.132 overlay (Overlay)

This element specifies that other chart elements shall be allowed to overlap this chart element.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.133 pageMargins (Page Margins)

This element specifies the page margins for a chart.

Attributes	Description
b (Bottom)	Specifies the bottom page margin in inches. The possible values for this attribute are defined by the W3C XML Schema double datatype.
footer (Footer)	Specifies the footer margin in inches. The possible values for this attribute are defined by the W3C XML Schema double datatype.
header (Header)	Specifies the header margin in inches. The possible values for this attribute are defined by the W3C XML Schema double datatype.
l (Left)	Specifies the left page margin in inches. The possible values for this attribute are defined by the W3C XML Schema double datatype.
r (Right)	Specifies the right page margin in inches. The possible values for this attribute are defined by the W3C XML Schema double datatype.
t (Top)	Specifies the top page margin in inches. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PageMargins) is located in §A.5.1. *end note*]

21.2.2.134 pageSetup (Page Setup)

This element defines the page setup for the chart.

[*Example*: The following example shows the pageSetup element for ISO A0 paper, printed in black and white, with graphics:

```
<pageSetup blackAndWhite="true" draft="false" paperHeight="1189mm"
paperWidth="841mm"/>
```

end example]

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>blackAndWhite (Black and White)</td><td>Specifies the page shall print in black and white.

The possible values for this attribute are defined by the W3C XML Schema boolean datatype.</td></tr>
<tr><td>copies (Copies)</td><td>Specifies the number of copies that shall be printed.

The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.</td></tr>
<tr><td>draft (Draft)</td><td>Specifies the page shall be printed in draft mode.

The possible values for this attribute are defined by the W3C XML Schema boolean datatype.</td></tr>
<tr><td>firstPageNumber (First Page Number)</td><td>Specifies the page number.

The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.</td></tr>
<tr><td>horizontalDpi (Horizontal DPI)</td><td>Specifies the horizontal resolution to print in dots per inch.

The possible values for this attribute are defined by the W3C XML Schema int datatype.</td></tr>
<tr><td>orientation (Orientation)</td><td>Specifies the orientation of the paper.

The possible values for this attribute are defined by the ST_PageSetupOrientation simple type (§21.2.3.32).</td></tr>
<tr><td>paperHeight (Paper Height)</td><td>Height of custom paper as a number followed by a unit identifier. [Example: <code>297mm</code>, <code>11in</code> end example]

When paperHeight and paperWidth are specified, paperSize shall be ignored.

The possible values for this attribute are defined by the ST_PositiveUniversalMeasure simple type (§22.9.2.12).</td></tr>
<tr><td>paperSize (Paper Size)</td><td>Specifies the paper size according to the following table.
<table>
<tr><th>Paper Size</th><th>Width</th><th>Height</th><th>Value</th></tr>
<tr><td>Letter paper</td><td>8.5 in.</td><td>11 in.</td><td>1</td></tr>
<tr><td>Legal paper</td><td>8.5 in.</td><td>14 in.</td><td>5</td></tr>
<tr><td>Standard paper</td><td>10 in.</td><td>11 in.</td><td>45</td></tr>
<tr><td>Standard paper</td><td>10 in.</td><td>14 in.</td><td>16</td></tr>
<tr><td>Standard paper</td><td>11 in.</td><td>17 in.</td><td>17</td></tr>
<tr><td>Standard paper</td><td>15 in.</td><td>11 in.</td><td>46</td></tr>
<tr><td>Standard paper</td><td>9 in.</td><td>11 in.</td><td>44</td></tr>
</table>
</td></tr>
</table>

Attributes	Description

SuperA/SuperA/A4 paper	227 mm	356 mm	57
A2 paper	420 mm	594 mm	66
A3 paper	297 mm	420 mm	8
A3 extra paper	322 mm	445 mm	63
A3 extra transverse paper	322 mm	445 mm	68
A3 transverse paper	297 mm	420 mm	67
A4 paper	210 mm	297 mm	9
A4 extra paper	236 mm	322 mm	53
A4 plus paper	210 mm	330 mm	60
A4 transverse paper	210 mm	297 mm	55
A4 small paper	210 mm	297 mm	10
A5 paper	148 mm	210 mm	11
A5 extra paper	174 mm	235 mm	64
A5 transverse paper	148 mm	210 mm	61
SuperB/SuperB/A3 paper	305 mm	487 mm	58
B4 paper	250 mm	353 mm	12
B5 paper	176 mm	250 mm	13
ISO B5 extra paper	201 mm	276 mm	65
JIS B5 transverse paper	182 mm	257 mm	62
C paper	17 in.	22 in.	24
D paper	22 in.	34 in.	25
#10 envelope	4.125 in.	9.5 in.	20
#11 envelope	4.5 in.	10.375 in.	21
#12 envelope	4.75 in.	11 in.	22
#14 envelope	5 in.	11.5 in.	23
#9 envelope	3.875 in.	8.875 in.	19
B4 envelope	250 mm	353 mm	33
B5 envelope	176 mm	250 mm	34
B6 envelope	176 mm	125 mm	35
C3 envelope	324 mm	458 mm	29
C4 envelope	229 mm	324 mm	30

Attributes	Description

C5 envelope	162 mm	229 mm	28
C6 envelope	114 mm	162 mm	31
C65 envelope	114 mm	229 mm	32
DL envelope	110 mm	220 mm	27
Invite envelope	220 mm	220 mm	47
Italy envelope	110 mm	230 mm	36
Monarch envelope	3.875 in.	7.5 in.).	37
6 3/4 envelope	3.625 in.	6.5 in.	38
E paper	34 in.	44 in.	26
Executive paper	7.25 in.	10.5 in.	7
German legal fanfold	8.5 in.	13 in.	41
German standard fanfold	8.5 in.	12 in.	40
US standard fanfold	14.875 in.	11 in.	39
Folio paper	8.5 in.	13 in.	14
ISO B4	250 mm	353 mm	42
Japanese double postcard	200 mm	148 mm	43
Ledger paper	17 in.	11 in.	4
Legal extra paper	9.275 in.	15 in.	51
Letter extra paper	9.275 in.	12 in.	50
Letter extra transverse paper	9.275 in.	12 in.	56
Letter plus paper	8.5 in.	12.69 in.	59
Letter transverse paper	8.275 in.	11 in.	54
Letter small paper	8.5 in.	11 in.	2
Note paper	8.5 in.	11 in.	18
Quarto paper	215 mm	275 mm	15
Statement paper	5.5 in.	8.5 in.	6
Tabloid paper	11 in.	17 in.	3
Tabloid extra paper	11.69 in.	18 in.	52

When paperHeight and paperWidth are specified, paperSize should be ignored.

The possible values for this attribute are defined by the W3C XML Schema unsignedInt

Attributes	Description
	datatype.
paperWidth (Paper Width)	Width of custom paper as a number followed by a unit identifier. [*Example*: `21cm`, `8.5in` *end example*] When paperHeight and paperWidth are specified, paperSize shall be ignored. The possible values for this attribute are defined by the ST_PositiveUniversalMeasure simple type (§22.9.2.12).
useFirstPageNumber (Use First Page Number)	Specifies to use the first page number instead of automatically generating a page number. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
verticalDpi (Vertical DPI)	Specifies the vertical resolution to print in dots per inch. 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_PageSetup) is located in §A.5.1. *end note*]

21.2.2.135 period (Period)

This element specifies the period of the trend line for a moving average trend line. It is ignored for other trend line variants.

Attributes	Description
val (Period Value)	Specifies the period of the trend line for a moving average trend line. Shall contain an integer between 2 and 255. The possible values for this attribute are defined by the ST_Period simple type (§21.2.3.33).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Period) is located in §A.5.1. *end note*]

21.2.2.136 perspective (Perspective)

This element specifies the field of view angle for the 3-D chart. This element is ignored if Right Angle Axes is true.

Attributes	Description
val (Perspective	Specifies the field of view angle for the 3-D chart. Shall contain an integer between 0

Attributes	Description
Value)	and 240, whose unit is one-half degrees. The possible values for this attribute are defined by the ST_Perspective simple type (§21.2.3.34).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Perspective) is located in §A.5.1. *end note*]

21.2.2.137 pictureFormat (Picture Format)

This element specifies the stretching and stacking of the picture on the data point, series, wall, or floor.

Attributes	Description
val (Picture Format Value)	Specifies the stretching and stacking of the picture. The possible values for this attribute are defined by the ST_PictureFormat simple type (§21.2.3.35).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PictureFormat) is located in §A.5.1. *end note*]

21.2.2.138 pictureOptions (Picture Options)

This element specifies the picture to be used on the data point, series, wall, or floor.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PictureOptions) is located in §A.5.1. *end note*]

21.2.2.139 pictureStackUnit (Picture Stack Unit)

This element specifies the unit for each picture on the chart. This element applies only if the Picture Format is Stack and Scale.

Attributes	Description
val (Picture Stack Unit)	Specifies the unit for each picture on the chart. Shall contain a floating point number. The possible values for this attribute are defined by the ST_PictureStackUnit simple type (§21.2.3.36).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PictureStackUnit) is located in §A.5.1. *end note*]

21.2.2.140 pie3DChart (3D Pie Charts)

This element contains the 3-D pie series for this chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Pie3DChart) is located in §A.5.1. *end note*]

21.2.2.141 pieChart (Pie Charts)

This element contains the 2-D pie series for this chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PieChart) is located in §A.5.1. *end note*]

21.2.2.142 pivotFmt (Pivot Format)

This element contains a set of formatting to be applied to the chart that is based on a pivotTable.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotFmt) is located in §A.5.1. *end note*]

21.2.2.143 pivotFmts (Pivot Formats)

This element contains a collection of formatting bands for a surface chart indexed from low to high.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotFmts) is located in §A.5.1. *end note*]

21.2.2.144 pivotSource (Pivot Source)

This element specifies the source pivot table for a pivot chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotSource) is located in §A.5.1. *end note*]

21.2.2.145 plotArea (Plot Area)

This element specifies the plot area of the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PlotArea) is located in §A.5.1. *end note*]

21.2.2.146 plotVisOnly (Plot Visible Only)

This element specifies that only visible cells should be plotted on the chart.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is

Attributes	Description
	omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.147 plus (Plus)

This element specifies the error bar value in the positive direction. It shall be used only when the errValType is cust.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumDataSource) is located in §A.5.1. *end note*]

21.2.2.148 printSettings (Print Settings)

This element specifies the print settings for the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PrintSettings) is located in §A.5.1. *end note*]

21.2.2.149 protection (Protection)

This element specifies protection for the chart. If the chart is on a protected worksheet or chart sheet, then these settings shall control how a user is able to interact with the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Protection) is located in §A.5.1. *end note*]

21.2.2.150 pt (Numeric Point)

This element specifies data for a particular data point.

Attributes	Description
formatCode (Number Format)	A string representing the format code to apply. For more information see see the SpreadsheetML numFmt element's (§18.8.30) formatCode attribute. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
idx (Index)	The index of the series in the collection The possible values for this attribute are defined by the W3C XML Schema unsignedInt

Attributes	Description
	datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumVal) is located in §A.5.1. *end note*]

21.2.2.151 pt (String Point)

This element specifies string data for a specific data point.

Attributes	Description
idx (Index)	A 0 based index into a set of points. Represents the data point number this data is for. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StrVal) is located in §A.5.1. *end note*]

21.2.2.152 ptCount (Point Count)

This element contains the number of values in the cache.

Attributes	Description
val (Integer Value)	Specifies that the contents of this attribute contain an integer number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnsignedInt) is located in §A.5.1. *end note*]

21.2.2.153 radarChart (Radar Charts)

This element contains the radar chart series on this chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_RadarChart) is located in §A.5.1. *end note*]

21.2.2.154 radarStyle (Radar Style)

This element specifies what type of radar chart shall be drawn.

Attributes	Description
val (Radar Style Value)	Specifies the style of the radar chart. The possible values for this attribute are defined by the ST_RadarStyle simple type (§21.2.3.37).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RadarStyle) is located in §A.5.1. *end note*]

21.2.2.155 rAngAx (Right Angle Axes)

This element specifies that the chart axes are at right angles, rather than drawn in perspective. Applies only to 3-D charts.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.156 rich (Rich Text)

This element contains a string with rich text formatting.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBody) is located in §A.4.1. *end note*]

21.2.2.157 rotX (X Rotation)

This element specifies the amount a 3-D chart shall be rotated in the X direction.

Attributes	Description
val (X Rotation Value)	Specifies the amount a 3-D chart shall be rotated in the X direction. Shall contain an integer between -90 and 90. The possible values for this attribute are defined by the ST_RotX simple type

Attributes	Description
	(§21.2.3.38).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RotX) is located in §A.5.1. *end note*]

21.2.2.158 rotY (Y Rotation)

This element specifies the amount a 3-D chart shall be rotated in the Y direction.

Attributes	Description
val (Y Rotation Value)	Specifies the amount a 3-D chart shall be rotated in the Y direction. Shall contain an integer between 0 and 360. The possible values for this attribute are defined by the ST_RotY simple type (§21.2.3.39).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RotY) is located in §A.5.1. *end note*]

21.2.2.159 roundedCorners (Rounded Corners)

This element specifies the chart area shall have rounded corners.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.160 scaling (Scaling)

This element contains additional axis settings.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Scaling) is located in §A.5.1. *end note*]

21.2.2.161 scatterChart (Scatter Charts)

This element contains the scatter chart series for this chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ScatterChart) is located in §A.5.1. *end note*]

21.2.2.162 scatterStyle (Scatter Style)

This element specifies the kind of lines for the scatter chart.

Attributes	Description
val (Scatter Style Value)	Specifies the style of the scatter chart. The possible values for this attribute are defined by the ST_ScatterStyle simple type (§21.2.3.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ScatterStyle) is located in §A.5.1. *end note*]

21.2.2.163 secondPiePt (Second Pie Point)

This element specifies a data point that shall be drawn in the second pie or bar in a pie of pie or bar of pie chart.

Attributes	Description
val (Integer Value)	Specifies that the contents of this attribute contain an integer number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnsignedInt) is located in §A.5.1. *end note*]

21.2.2.164 secondPieSize (Second Pie Size)

This element specifies the size of the second pie or bar of a pie of pie chart or a bar of pie chart.

Attributes	Description
val (Second Pie Size Value)	Specifies the second pie or bar of a pie of pie chart or a bar of pie chart, as a percentage of the size of the first pie. Shall contain a percentage between 5% and 200%. The possible values for this attribute are defined by the ST_SecondPieSize simple type (§21.2.3.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SecondPieSize) is located in §A.5.1. *end note*]

21.2.2.165 selection (Selection)

This element specifies the chart elements are protected from selection.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.166 separator (Separator)

This element specifies text that shall be used to separate the parts of a data label. The default is a comma, except for pie charts showing only category name and percentage, when a line break shall be used instead.

The possible values for this element are defined by the W3C XML Schema string datatype.

21.2.2.167 ser (Scatter Chart Series)

This element specifies a series on a scatter chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ScatterSer) is located in §A.5.1. *end note*]

21.2.2.168 ser (Area Chart Series)

This element specifies a series on an area chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AreaSer) is located in §A.5.1. *end note*]

21.2.2.169 ser (Radar Chart Series)

This element specifies a series on a radar chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_RadarSer) is located in §A.5.1. *end note*]

21.2.2.170 ser (Bar Chart Series)

This element specifies a series on a bar chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BarSer) is located in §A.5.1. *end note*]

21.2.2.171 ser (Line Chart Series)

This element specifies a series on a line chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineSer) is located in §A.5.1. *end note*]

21.2.2.172 ser (Pie Chart Series)

This element specifies a series on a doughnut or pie chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PieSer) is located in §A.5.1. *end note*]

21.2.2.173 ser (Surface Chart Series)

This element specifies a series on a surface chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SurfaceSer) is located in §A.5.1. *end note*]

21.2.2.174 ser (Bubble Chart Series)

This element specifies a series on a bubble chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BubbleSer) is located in §A.5.1. *end note*]

21.2.2.175 serAx (Series Axis)

This element specifies a series axis for the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SerAx) is located in §A.5.1. *end note*]

21.2.2.176 serLines (Series Lines)

This element specifies series lines for the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChartLines) is located in §A.5.1. *end note*]

21.2.2.177 shape (Shape)

This element specifies the shape of a series or a 3-D bar chart.

Attributes	Description
val (Shape Value)	Specifies the shape of the series. The possible values for this attribute are defined by the ST_Shape simple type (§21.2.3.42).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Shape) is located in §A.5.1. *end note*]

21.2.2.178 showBubbleSize (Show Bubble Size)

This element specifies the bubble size shall be shown in a data label.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.179 showCatName (Show Category Name)

This element specifies that the category name shall be shown in the data label.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied.

Attributes	Description
	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_Boolean) is located in §A.5.1. *end note*]

21.2.2.180 showDLblsOverMax (Show Data Labels over Maximum)

This element specifies data labels over the maximum of the chart shall be shown.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.181 showHorzBorder (Show Horizontal Border)

This element specifies the horizontal borders shall be shown in a data table.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.182 showKeys (Show Legend Keys)

This element specifies the legend keys shall be shown in a data table.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.183 showLeaderLines (Show Leader Lines)

This element specifies leader lines shall be shown for data labels.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.184 showLegendKey (Show Legend Key)

This element specifies legend keys shall be shown in data labels.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element.

Attributes	Description
	A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.185 showNegBubbles (Show Negative Bubbles)

This element specifies negative sized bubbles shall be shown on a bubble chart.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.186 showOutline (Show Outline Border)

This element specifies the outline shall be shown on a data table.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied.

Attributes	Description
	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_Boolean) is located in §A.5.1. *end note*]

21.2.2.187 showPercent (Show Percent)

This element specifies that the percentage shall be shown in a data label.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.188 showSerName (Show Series Name)

This element specifies that the series name shall be shown in a data label.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.189 showVal (Show Value)

This element specifies that the value shall be shown in a data label.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.190 showVertBorder (Show Vertical Border)

This element specifies the vertical border shall be shown in a data table.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.191 sideWall (Side Wall)

This element specifies the side wall.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Surface) is located in §A.5.1. *end note*]

21.2.2.192 size (Size)

This element specifies the size of the marker in points.

Attributes	Description
val (Marker Size Value)	Specifies the size of the marker in points. Shall contain an integer between 2 and 72. The possible values for this attribute are defined by the ST_MarkerSize simple type (§21.2.3.26).

[*Note*: The W3C XML Schema definition of this element's content model (CT_MarkerSize) is located in §A.5.1. *end note*]

21.2.2.193 sizeRepresents (Size Represents)

This element specifies how the bubble size values are represented on the chart.

Attributes	Description
val (Size Represents Value)	Specifies how the bubble sizes represent the values. The possible values for this attribute are defined by the ST_SizeRepresents simple type (§21.2.3.43).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SizeRepresents) is located in §A.5.1. *end note*]

21.2.2.194 smooth (Smoothing)

This element specifies the line connecting the points on the chart shall be smoothed using Catmull-Rom splines.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.195 splitPos (Split Position)

This element specifies a value that shall be used to determine which data points are in the second pie or bar on a pie of pie or bar of pie chart.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.196 splitType (Split Type)

This element specifies how to determine which data points are in the second pie or bar on a pie of pie or bar of pie chart.

Attributes	Description
val (Split Type Value)	Specifies how to split the data points between the first pie and second pie or bar. The possible values for this attribute are defined by the ST_SplitType simple type (§21.2.3.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SplitType) is located in §A.5.1. *end note*]

21.2.2.197 spPr (Shape Properties)

This element specifies the formatting for the parent chart element. The custGeom, prstGeom, scene3d, and xfrm elements are not supported. The bwMode attribute is not supported.

Attributes	Description
bwMode (Black and White Mode) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeProperties) is located in §A.4.1. *end note*]

21.2.2.198 stockChart (Stock Charts)

This element contains the collection of stock chart series.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StockChart) is located in §A.5.1. *end note*]

21.2.2.199 strCache (String Cache)

This element specifies the last string data used for a chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StrData) is located in §A.5.1. *end note*]

21.2.2.200 strLit (String Literal)

This element specifies a set of strings used for a chart

[*Note*: The W3C XML Schema definition of this element's content model (CT_StrData) is located in §A.5.1. *end note*]

21.2.2.201 strRef (String Reference)

This element specifies a reference to data for a single data label or title with a cache of the last values used.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StrRef) is located in §A.5.1. *end note*]

21.2.2.202 style (Style)

This element specifies the style that shall be applied to the chart.

Attributes	Description
val (Style Type)	Specifies the chart style. The possible values for this attribute are defined by the ST_Style simple type (§21.2.3.46).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Style) is located in §A.5.1. *end note*]

21.2.2.203 surface3DChart (3D Surface Charts)

This element contains the set of 3-D surface series.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Surface3DChart) is located in §A.5.1. *end note*]

21.2.2.204 surfaceChart (Surface Charts)

This element contains the set of 2-D contour charts.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SurfaceChart) is located in §A.5.1. *end note*]

21.2.2.205 symbol (Symbol)

This element specifies the marker that is used for the data points.

Attributes	Description
val (Marker Style Value)	Specifies the marker style. The possible values for this attribute are defined by the ST_MarkerStyle simple type (§21.2.3.27).

[*Note*: The W3C XML Schema definition of this element's content model (CT_MarkerStyle) is located in §A.5.1. *end note*]

21.2.2.206 thickness (Thickness)

This element specifies the thickness of the walls or floor as a percentage of the largest dimension of the plot volume.

Attributes	Description
val (Integer Value)	Specifies that the contents of this attribute contain a percentage. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the ST_Thickness simple type (§21.2.3.59).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Thickness) is located in §A.5.1. *end note*]

21.2.2.207 tickLblPos (Tick Label Position)

This element specifies the position of the tick labels on the axis.

Attributes	Description
val (Tick Label Position Value)	Specifies the tick label position. The possible values for this attribute are defined by the ST_TickLblPos simple type (§21.2.3.47).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TickLblPos) is located in §A.5.1. *end note*]

21.2.2.208 tickLblSkip (Tick Label Skip)

This element specifies how many tick labels to skip between label that is drawn.

Attributes	Description
val (Tick Skip Value)	Specifies the how many tick labels to skip between label that is drawn. Shall contain an integer greater than or equal to one. The possible values for this attribute are defined by the ST_Skip simple type (§21.2.3.44).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Skip) is located in §A.5.1. *end note*]

21.2.2.209 tickMarkSkip (Tick Mark Skip)

This element specifies how many tick marks shall be skipped before the next one shall be drawn.

Attributes	Description
val (Tick Skip Value)	Specifies the how many tick marks shall be skipped before the next one shall be drawn. Shall contain an integer greater than or equal to one. The possible values for this attribute are defined by the ST_Skip simple type (§21.2.3.44).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Skip) is located in §A.5.1. *end note*]

21.2.2.210 title (Title)

This element specifies a title.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Title) is located in §A.5.1. *end note*]

21.2.2.211 trendline (Trendlines)

This element specifies a trendline.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Trendline) is located in §A.5.1. *end note*]

21.2.2.212 trendlineLbl (Trendline Label)

This element specifies the label for the trendline.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrendlineLbl) is located in §A.5.1. *end note*]

21.2.2.213 trendlineType (Trendline Type)

This element specifies the style of the trendline.

Attributes	Description
val (Trendline Type Value)	Specifies the trendline style. The possible values for this attribute are defined by the ST_TrendlineType simple type (§21.2.3.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrendlineType) is located in §A.5.1. *end note*]

21.2.2.214 tx (Chart Text)

This element specifies text to use on a chart, including rich text formatting.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Tx) is located in §A.5.1. *end note*]

21.2.2.215 tx (Series Text)

This element specifies text for a series name, without rich text formatting.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SerTx) is located in §A.5.1. *end note*]

21.2.2.216 txPr (Text Properties)

This element specifies text formatting. The lstStyle element is not supported.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBody) is located in §A.4.1. *end note*]

21.2.2.217 upBars (Up Bars)

This element specifies the up bars on the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UpDownBar) is located in §A.5.1. *end note*]

21.2.2.218 upDownBars (Up/Down Bars)

This element specifies the up and down bars.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UpDownBars) is located in §A.5.1. *end note*]

21.2.2.219 userInterface (User Interface)

This element specifies that the protection applies to the user interface only, and not to changes made through the object model.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.220 userShapes (User Shapes)

This element shall specify the shapes drawn on top of the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Drawing) is located in §A.5.2. *end note*]

21.2.2.221 userShapes (Reference to Chart Drawing Part)

This element specifies a relationship to a separate part which contains a drawing to be drawn on top of the chart.

Attributes	Description
id (Relationship Reference) Namespace: ttp://purl.oclc.org/ooxml/officeDocument/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 (§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*]

21.2.2.222 v (Numeric Value)

This element specifies a numeric value.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

21.2.2.223 v (Text Value)

This element specifies a text value for a category axis label or a series name.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

21.2.2.224 val (Values)

This element specifies the data values which shall be used to define the location of data markers on a chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumDataSource) is located in §A.5.1. *end note*]

21.2.2.225 val (Error Bar Value)

This element specifies a value which is used with the errBar element to determine the length of the error bars.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.226 valAx (Value Axis)

This element specifies a value axis.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ValAx) is located in §A.5.1. *end note*]

21.2.2.227 varyColors (Vary Colors by Point)

This element specifies that each data marker in the series has a different color.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.228 view3D (View In 3D)

This element specifies the 3-D view of the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_View3D) is located in §A.5.1. *end note*]

21.2.2.229 w (Width)

This element specifies the width (if Width Mode is Factor) or right (if Width Mode is Edge) of the chart element as a fraction of the width of the chart.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.230 wireframe (Wireframe)

This element specifies the surface chart is drawn as a wireframe.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. 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_Boolean) is located in §A.5.1. *end note*]

21.2.2.231 wMode (Width Mode)

This element specifies how to interpret the Width element for this manual layout.

Attributes	Description
val (Layout Mode Value)	Specifies the layout mode for the width. The possible values for this attribute are defined by the ST_LayoutMode simple type (§21.2.3.20).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LayoutMode) is located in §A.5.1. *end note*]

21.2.2.232 x (Left)

This element specifies the x location (left) of the chart element as a fraction of the width of the chart. If Left Mode is Factor, then the position is relative to the default position for the chart element.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.233 xMode (Left Mode)

This element specifies how to interpret the Left element for this manual layout.

Attributes	Description
val (Layout Mode Value)	Specifies the layout mode for the width. The possible values for this attribute are defined by the ST_LayoutMode simple type (§21.2.3.20).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LayoutMode) is located in §A.5.1. *end note*]

21.2.2.234 xVal (X Values)

This element specifies the x values which shall be used to define the location of data markers on a chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AxDataSource) is located in §A.5.1. *end note*]

21.2.2.235 y (Top)

This element specifies the top of the chart element as a fraction of the height of the chart. If Top Mode is Factor, then the position is relative to the default position for the chart element.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.236 yMode (Top Mode)

This element specifies how to interpret the Top element for this manual layout.

Attributes	Description
val (Layout Mode Value)	Specifies the layout mode for the width. The possible values for this attribute are defined by the ST_LayoutMode simple type (§21.2.3.20).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LayoutMode) is located in §A.5.1. *end note*]

21.2.2.237 yVal (Y Values)

This element specifies the y values which shall be used to define the location of data markers on a chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumDataSource) is located in §A.5.1. *end note*]

21.2.3 Simple Types

This is the complete list of simple types dedicated to DrawingML – Charts.

21.2.3.1 ST_AxisUnit (Axis Unit)

This simple type specifies that its contents contain a positive floating point number.

This simple type's contents are a restriction of the W3C XML Schema double datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than 0.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AxisUnit) is located in §A.5.1. *end note*]

21.2.3.2 ST_AxPos (Axis Position)

This simple type specifies the possible positions for an axis.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bottom)	Specifies that the axis shall be displayed at the bottom of the plot area.
l (Left)	Specifies that the axis shall be displayed at the left of the plot area.
r (Right)	Specifies that the axis shall be displayed at the right of the plot area.

Enumeration Value	Description
t (Top)	Specifies that the axis shall be displayed at the top of the plot area.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AxPos) is located in §A.5.1. *end note*]

21.2.3.3 ST_BarDir (Bar Direction)

This simple type specifies the possible directions for a bar chart.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bar (Bar)	Specifies that the chart is a bar chart - the data markers are horizontal rectangles.
col (Column)	Specifies that the chart is a column chart - the data markers are vertical rectangles.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BarDir) is located in §A.5.1. *end note*]

21.2.3.4 ST_BarGrouping (Bar Grouping)

This simple type specifies the possible groupings for a bar chart.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
clustered (Clustered)	Specifies that the chart series are drawn next to each other along the category axis.
percentStacked (100% Stacked)	Specifies that the chart series are drawn next to each other along the value axis and scaled to total 100%.
stacked (Stacked)	Specifies that the chart series are drawn next to each other on the value axis.
standard (Standard)	Specifies that the chart series are drawn next to each other on the depth axis.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BarGrouping) is located in §A.5.1. *end note*]

21.2.3.5 ST_BubbleScale (Bubble Scale)

This simple type specifies that its contents contain a percentage between 0% and 300%.

This simple type is a union of the following types:

- ST_BubbleScalePercent simple type (§21.2.3.58).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BubbleScale) is located in §A.5.1. *end note*]

21.2.3.6 ST_BuiltInUnit (Built-In Unit)

This simple type specifies the built in display units for an axis.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
billions (Billions)	Specifies the values on the chart shall be divided by 1,000,000,000.
hundredMillions (Hundred Millions)	Specifies the values on the chart shall be divided by 100,000,000.
hundreds (Hundreds)	Specifies the values on the chart shall be divided by 100.
hundredThousands (Hundred Thousands)	Specifies the values on the chart shall be divided by 100,000.
millions (Millions)	Specifies the values on the chart shall be divided by 1,000,000.
tenMillions (Ten Millions)	Specifies the values on the chart shall be divided by 10,000,000.
tenThousands (Ten Thousands)	Specifies the values on the chart shall be divided by 10,000.
thousands (Thousands)	Specifies the values on the chart shall be divided by 1,000.
trillions (Trillions)	Specifies the values on the chart shall be divided by 1,000,000,000.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BuiltInUnit) is located in §A.5.1. *end note*]

21.2.3.7 ST_CrossBetween (Cross Between)

This simple type specifies the possible crossing states of an axis.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
between (Between)	Specifies the value axis shall cross the category axis between data markers.
midCat (Midpoint of Category)	Specifies the value axis shall cross the category axis at the midpoint of a category.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CrossBetween) is located in §A.5.1. *end note*]

21.2.3.8 ST_Crosses (Crosses)

This simple type specifies the possible crossing points for an axis.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
autoZero (Axis Crosses at Zero)	The category axis crosses at the zero point of the value axis (if possible), or the minimum value (if the minimum is greater than zero) or the maximum (if the maximum is less than zero).
max (Maximum)	The axis crosses at the maximum value
min (Minimum)	Axis crosses at the minimum value of the chart.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Crosses) is located in §A.5.1. *end note*]

21.2.3.9 ST_DepthPercent (Depth Percent)

This simple type specifies that its contents contain a percentage between 20% and 2000%.

This simple type is a union of the following types:

- ST_DepthPercentWithSymbol simple type (§21.2.3.51).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DepthPercent) is located in §A.5.1. *end note*]

21.2.3.10 ST_DispBlanksAs (Display Blanks As)

This simple type specifies the possible ways to display blanks.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
gap (Gap)	Specifies that blank values shall be left as a gap.
span (Span)	Specifies that blank values shall be spanned with a line.
zero (Zero)	Specifies that blank values shall be treated as zero.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DispBlanksAs) is located in §A.5.1. *end note*]

21.2.3.11 ST_DLblPos (Data Label Position)

This simple type specifies the possible positions for a data label.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bottom)	Specifies that data labels shall be displayed below the data marker.
bestFit (Best Fit)	Specifies that data labels shall be displayed in the best position.
ctr (Center)	Specifies that data labels shall be displayed centered on the data marker.
inBase (Inside Base)	Specifies that data labels shall be displayed inside the base of the data marker.
inEnd (Inside End)	Specifies that data labels shall be displayed inside the end of the data marker.
l (Left)	Specifies that data labels shall be displayed to the left of the data marker.
outEnd (Outside End)	Specifies that data labels shall be displayed outside the end of the data marker.
r (Right)	Specifies that data labels shall be displayed to the right of the data marker.
t (Top)	Specifies that data labels shall be displayed above the data marker.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DLblPos) is located in §A.5.1. *end note*]

21.2.3.12 ST_ErrBarType (Error Bar Type)

This simple type specifies the possible ways to draw an error bar.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
both (Both)	Specifies that error bars shall be shown in the positive and negative directions.
minus (Minus)	Specifies that error bars shall be shown in the negative direction only.
plus (Plus)	Specifies that error bars shall be shown in the positive direction only.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ErrBarType) is located in §A.5.1. *end note*]

21.2.3.13 ST_ErrDir (Error Bar Direction)

This simple type specifies the possible directions for error bars.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
x (X)	Specifies that error bars shall be shown in the x direction.
y (Y)	Specifies that error bars shall be shown in the y direction.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ErrDir) is located in §A.5.1. *end note*]

21.2.3.14 ST_ErrValType (Error Value Type)

This simple type specifies the possible ways to determine the length of the error bars

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
cust (Custom Error Bars)	Specifies that the length of the error bars shall be determined by the Plus and Minus elements.

Enumeration Value	Description
fixedVal (Fixed Value)	Specifies that the length of the error bars shall be the fixed value determined by Error Bar Value.
percentage (Percentage)	Specifies that the length of the error bars shall be Error Bar Value percent of the data.
stdDev (Standard Deviation)	Specifies that the length of the error bars shall be Error Bar Value standard deviations of the data.
stdErr (Standard Error)	Specifies that the length of the error bars shall be Error Bar Value standard errors of the data.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ErrValType) is located in §A.5.1. *end note*]

21.2.3.15 ST_FirstSliceAng (First Slice Angle)

This simple type specifies that its contents contain an integer between 0 and 360.

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 360.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FirstSliceAng) is located in §A.5.1. *end note*]

21.2.3.16 ST_GapAmount (Gap Amount)

This simple type specifies that its contents contain a percentage between 0% and 500%.

This simple type is a union of the following types:

- ST_GapAmountPercent simple type (§21.2.3.53).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_GapAmount) is located in §A.5.1. *end note*]

21.2.3.17 ST_Grouping (Grouping)

This simple type specifies the possible groupings for a bar chart.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
percentStacked (100% Stacked)	Specifies that the chart series are drawn next to each other along the value axis and scaled to total 100%.
stacked (Stacked)	Specifies that the chart series are drawn next to each other on the value axis.
standard (Standard)	Specifies that the chart series are drawn on the value axis.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Grouping) is located in §A.5.1. *end note*]

21.2.3.18 ST_HoleSize (Hole Size)

This simple type specifies that its contents contain a percentage between 1% and 90%.

This simple type is a union of the following types:

- ST_HoleSizePercent simple type (§21.2.3.55).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HoleSize) is located in §A.5.1. *end note*]

21.2.3.19 ST_HPercent (Height Percent)

This simple type specifies that its contents contain a percentage between 5% and 500%.

This simple type is a union of the following types:

- ST_HPercentWithSymbol simple type (§21.2.3.52).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HPercent) is located in §A.5.1. *end note*]

21.2.3.20 ST_LayoutMode (Layout Mode)

This simple type specifies the possible ways to store a chart element's position.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
edge (Edge)	Specifies that the Width or Height shall be interpreted as the Right or Bottom of the chart element.
factor (Factor)	Specifies that the Width or Height shall be interpreted as the Width or Height of the chart element.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LayoutMode) is located in §A.5.1. *end note*]

21.2.3.21 ST_LayoutTarget (Layout Target)

This simple type specifies the possible ways to layout the plot area.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
inner (Inner)	Specifies that the plot area size shall determine the size of the plot area, not including the tick marks and axis labels.
outer (Outer)	Specifies that the plot area size shall determine the size of the plot area, the tick marks, and the axis labels.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LayoutTarget) is located in §A.5.1. *end note*]

21.2.3.22 ST_LblAlgn (Label Alignment)

This simple type specifies the possible ways to align the tick labels.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ctr (Center)	Specifies that the text shall be centered.
l (Left)	Specifies that the text shall be left justified.
r (Right)	Specifies that the text shall be right justified.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LblAlgn) is located in §A.5.1. *end note*]

21.2.3.23 ST_LblOffset (Label Offset)

This simple type specifies that its contents contain a percentage of the default value, between 0% and 1000%.

This simple type is a union of the following types:

- ST_LblOffsetPercent simple type (§21.2.3.56).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LblOffset) is located in §A.5.1. *end note*]

21.2.3.24 ST_LegendPos (Legend Position)

This simple type specifies the possible positions for a legend.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bottom)	Specifies that the legend shall be drawn at the bottom of the chart.
l (Left)	Specifies that the legend shall be drawn at the left of the chart.
r (Right)	Specifies that the legend shall be drawn at the right of the chart.
t (Top)	Specifies that the legend shall be drawn at the top of the chart.
tr (Top Right)	Specifies that the legend shall be drawn at the top right of the chart.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LegendPos) is located in §A.5.1. *end note*]

21.2.3.25 ST_LogBase (Logarithmic Base)

This simple type specifies that its contents contain a floating point number greater than or equal to two.

This simple type's contents are a restriction of the W3C XML Schema double datatype.

This simple type also specifies the following restrictions:

- This simple type has a maximum value of less than or equal to 1000.
- This simple type has a minimum value of greater than or equal to 2.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LogBase) is located in §A.5.1. *end note*]

21.2.3.26 ST_MarkerSize (Marker Size)

This simple type specifies that its contents contain an integer between 2 and 72, whose contents are a size in points.

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 2.
- This simple type has a maximum value of less than or equal to 72.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_MarkerSize) is located in §A.5.1. *end note*]

21.2.3.27 ST_MarkerStyle (Marker Style)

This picture shows each of the marker styles. Black is used as the line color, while red is used as the fill color. The height of the dash and the dot are 1/5th of the height of the marker. The width of the dot is 1/2 the width of the marker. The dash and dot have fills as well, but the markers need to be made quite large before these are visible.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
circle (Circle)	Specifies a circle shall be drawn at each data point.
dash (Dash)	Specifies a dash shall be drawn at each data point.
diamond (Diamond)	Specifies a diamond shall be drawn at each data point.
dot (Dot)	Specifies a dot shall be drawn at each data point.
none (None)	Specifies nothing shall be drawn at each data point.
picture (Picture)	Specifies a picture shall be drawn at each data point.
plus (Plus)	Specifies a plus shall be drawn at each data point.
square (Square)	Specifies a square shall be drawn at each data point.
star (Star)	Specifies a star shall be drawn at each data point.
triangle (Triangle)	Specifies a triangle shall be drawn at each data point.
x (X)	Specifies an X shall be drawn at each data point.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_MarkerStyle) is located in §A.5.1. *end note*]

21.2.3.28 ST_OfPieType (Pie of Pie or Bar of Pie Type)

This simple type specifies the possible modes of Pie of Pie or Bar of Pie charts.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bar (Bar)	Specifies that the chart is a bar of pie chart, not a pie of pie chart.
pie (Pie)	Specifies that the chart is pie of pie chart, not a bar of pie chart.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_OfPieType) is located in §A.5.1. *end note*]

21.2.3.29 ST_Order (Order)

This simple type specifies that its contents contain an integer between 2 and 6, whose contents are the order of the trendline polynomial.

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 2.
- This simple type has a maximum value of less than or equal to 6.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Order) is located in §A.5.1. *end note*]

21.2.3.30 ST_Orientation (Orientation)

This simple type specifies the possible ways to place a picture on a data point, series, wall, or floor.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
maxMin (Maximum to Minimum)	Specifies that the values on the axis shall be reversed so they go from maximum to minimum.

Enumeration Value	Description
minMax (Minimum to Maximum)	Specifies that the axis values shall be in the usual order, minimum to maximum.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Orientation) is located in §A.5.1. *end note*]

21.2.3.31 ST_Overlap (Overlap)

This simple type specifies that its contents contain a percentage between -100% and 100%.

This simple type is a union of the following types:

- ST_OverlapPercent simple type (§21.2.3.57).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Overlap) is located in §A.5.1. *end note*]

21.2.3.32 ST_PageSetupOrientation (Printed Page Orientation)

This simple type specifies the page orientation of the printed page(s) on which this chart shall appear.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
default (Default Page Orientation)	Specifies that the page orientation shall be the default orientation of the system.
landscape (Landscape Page)	Specifies that the printed page shall have landscape orientation.
portrait (Portrait Page)	Specifies that the printed page shall have portrait orientation.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PageSetupOrientation) is located in §A.5.1. *end note*]

21.2.3.33 ST_Period (Period)

This simple type specifies that its contents contain an integer greater than or equal to 2.

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 2.

Enumeration Value	Description
auto (Auto)	Specifies an application-specific marker shall be drawn at each data point.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Period) is located in §A.5.1. *end note*]

21.2.3.34 ST_Perspective (Perspective)

This simple type specifies that its contents contain an integer between 0 and 100, 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 0.
- This simple type has a maximum value of less than or equal to 240.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Perspective) is located in §A.5.1. *end note*]

21.2.3.35 ST_PictureFormat (Picture Format)

This simple type specifies the possible ways to place a picture on a data point, series, wall, or floor.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
stack (Stack)	Specifies that the picture shall be stacked.
stackScale (Stack and Scale)	Specifies that the picture shall be stacked after being scaled so that it's height is one Picture Stack Unit. Does not apply to walls or floor.
stretch (Stretch)	Specifies that the picture shall be anisotropic stretched to fill the data point, series, wall or floor.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PictureFormat) is located in §A.5.1. *end note*]

21.2.3.36 ST_PictureStackUnit (Picture Stack Unit)

This simple type specifies that its contents contain a floating point number greater than zero.

This simple type's contents are a restriction of the W3C XML Schema double datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than 0.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PictureStackUnit) is located in §A.5.1. *end note*]

21.2.3.37 ST_RadarStyle (Radar Style)

This simple type specifies the possible styles of radar chart.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
filled (Filled)	Specifies that the radar chart shall be filled and have lines but no markers.
marker (Marker)	Specifies that the radar chart shall have lines and markers but no fill.
standard (Standard)	Specifies that the radar chart shall have lines but no markers and no fill.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RadarStyle) is located in §A.5.1. *end note*]

21.2.3.38 ST_RotX (X Rotation)

This simple type specifies that its contents contain an integer between -90 and 90, whose contents are an angle in degrees.

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 -90.
- This simple type has a maximum value of less than or equal to 90.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RotX) is located in §A.5.1. *end note*]

21.2.3.39 ST_RotY (Y Rotation)

This simple type specifies that its contents contain an integer between 0 and 360, whose contents are an angle in degrees.

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 360.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RotY) is located in §A.5.1. *end note*]

21.2.3.40 ST_ScatterStyle (Scatter Style)

This simple type specifies the possible styles of scatter chart.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
line (Line)	Specifies the points on the scatter chart shall be connected with straight lines but markers shall not be drawn.
lineMarker (Line with Markers)	Specifies the points on the scatter chart shall be connected with straight lines and markers shall be drawn.
marker (Marker)	Specifies the points on the scatter chart shall not be connected with lines and markers shall be drawn.
none (None)	Specifies the points on the scatter chart shall not be connected with straight lines and markers shall not be drawn.
smooth (Smooth)	Specifies the the points on the scatter chart shall be connected with smoothed lines and markers shall not be drawn.
smoothMarker (Smooth with Markers)	Specifies the the points on the scatter chart shall be connected with smoothed lines and markers shall be drawn.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ScatterStyle) is located in §A.5.1. *end note*]

21.2.3.41 ST_SecondPieSize (Second Pie Size)

This simple type specifies that its contents contain a percentage between 5% and 200%, whose contents consist of a percentage.

This simple type is a union of the following types:

- ST_SecondPieSizePercent simple type (§21.2.3.55).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SecondPieSize) is located in §A.5.1. *end note*]

21.2.3.42 ST_Shape (Shape)

This simple type specifies the possible shapes for a 3-D data marker.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
box (Box)	Specifies the chart shall be drawn with a box shape.
cone (Cone)	Specifies the chart shall be drawn as a cone, with the base of the cone on the floor and the point of the cone at the top of the data marker.
coneToMax (Cone to Max)	Specifies the chart shall be drawn with truncated cones such that the point of the cone would be the maximum data value.
cylinder (Cylinder)	Specifies the chart shall be drawn as a cylinder.
pyramid (Pyramid)	Specifies the chart shall be drawn as a rectangular pyramid, with the base of the pyramid on the floor and the point of the pyramid at the top of the data marker.
pyramidToMax (Pyramid to Maximum)	Specifies the chart shall be drawn with truncated cones such that the point of the cone would be the maximum data value.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Shape) is located in §A.5.1. *end note*]

21.2.3.43 ST_SizeRepresents (Size Represents)

This simple type specifies the possible ways to represent data as bubble chart sizes.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
area (Bubble Size Represents Area)	Specifies the area of the bubbles shall be proportional to the bubble size value.
w (Bubble Size Represents Width)	Specifies the radius of the bubbles shall be proportional to the bubble size value.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SizeRepresents) is located in §A.5.1. *end note*]

21.2.3.44 ST_Skip (Skip)

This simple type specifies that its contents contain an integer greater than or equal to one.

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 1.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Skip) is located in §A.5.1. *end note*]

21.2.3.45 ST_SplitType (Split Type)

This simple type specifies the possible ways to split a pie of pie or bar of pie chart.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
auto (Default Split)	Specifies the data points shall be split using the default mechanism for this chart type.
cust (Custom Split)	Specifies the data points shall be split between the pie and the second chart according to the Custom Split values.
percent (Split by Percentage)	Specifies the data points shall be split between the pie and the second chart by putting the points with percentage less than Split Position percent in the second chart.
pos (Split by Position)	Specifies the data points shall be split between the pie and the second chart by putting the last Split Position of the data points in the second chart
val (Split by Value)	Specifies the data points shall be split between the pie and the second chart by putting the data points with

Enumeration Value	Description
	value less than Split Position in the second chart

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SplitType) is located in §A.5.1. *end note*]

21.2.3.46 ST_Style (Style)

This simple type specifies that its contents contain an integer between 1 and 48. The value determines the default formatting for all chart elements through the tables described below.

The default font is the minor font as defined by the document's theme. The default font size for each element is the font size of the chart, except for the title which is always 120% the font size of the chart. If the chart does not have a font size set, then the default font size is 10. Axis titles and chart titles are bold by default, while all other chart elements are normal. The default font color is the same as the Axis & Major Gridlines Line Color.

The default line style, fill style, and effect style are determined by the tables below. Each of the default includes a themed line, fill, or effect (None, Subtle, Moderate, or Intense) and a color to be used when applying that line, fill, or effect. In some cases, both the themed formatting and the color vary per style, in other cases they do not. The default line width is determined by the theme, except for lines for data points it is multipled by the line width value given in the table.

Table 1: Chart element defaults

This table lists whether the default is constant or whether it depends on the ST_Style using one of the later tables.

	Line		Fill		Effect	
Chart Element	Themed Line	Color	Themed Fill	Color	Themed Effect	Color
Axis	Subtle	Table 2	No Fill		No Effect	
Axis Title	No Line		No Fill		No Effect	
Chart Area	Table 2		Subtle	Table 3	No Effect	
Chart Title	No Line		No Fill		No Effect	
Data Labels	No Line		No Fill		No Effect	
Data Table	Subtle	Table 2	No Fill		No Effect	
Down Bars	Table 4		Table 4		Table 4	dk1
Fills for Data Points (2-D)	Subtle	Table 5	Table 5		Table 5	dk1
Fills for Data Points (3-D)	Subtle	Table 5	Table 5		Table 5	dk1
Floor	Table 2		Table 3		No Effect	
Legend	No Line		No Fill		No Effect	
Lines for Data Points	Subtle	Table 5	N/A		No Effect	

	Line		Fill		Effect	
Major Gridlines	Subtle	Table 2	N/A		No Effect	
Markers for Data Points	Subtle	Table 5	Table 5		Table 5	dk1
Minor Gridlines	Subtle	Table 2	N/A		No Effect	
Other Lines	Subtle	Table 2	N/A		No Effect	
Plot Area (2-D charts)	No Line		Subtle	Table 3	No Effect	
Plot Area (3-D charts)	No Line		No Fill		No Effect	
Trendline Labels	No Line		No Fill		No Effect	
Up Bars	Table 4		Table 4		Table 4	dk1
Walls	No Line		Table 3		No Effect	

Other Lines includes Drop Lines, Error Bars, High Low Lines, Leader Lines, Series Lines, and Trendlines.

Fills for Data Points (2-D) includes 2-D bar, filled radar, stock, bubble, pie, doughnut and area charts.

Fills for Data Points (3-D) includes all 3-D charts.

Lines for Data Points includes lines on 2-D line, scatter, bubble, and radar charts.

Markers for Data Points includes markers on 2-D line, scatter, and radar charts.

Table 2: Default line formatting per chart style

This table lists line formatting for several chart elements by style.

Style	Axis & Major Gridlines	Minor Gridlines	Chart Area, Data Table, & Floor	Other Lines	Floor & Chart Area
	Color	Color	Color	Color	Themed Line
1-32	tx1	50% tint of tx1	75% tint of tx1	tx1	Subtle
33-34	dk1	50% tint of tx1	75% tint of dk1	dk1	Subtle
35-40	dk1	50% tint of tx1	75% tint of dk1	25% shade of dk1	Subtle
41-48	dk1	90% tint of tx1	lt1	lt1	No Line

Table 3: Default fill formatting per chart style

This table lists fill formatting for several chart elements by style.

Style	Chart Area	Floor, Walls & Plot Area (2-D)	Floor & Walls
	Color	Color	Themed Fill
1-32	bg1	bg1	No Fill
33-34	lt1	20% tint of dk1	Subtle
35-40	lt1	accent1-6	Subtle
41-48	dk1	95% tint of dk1	Subtle

Table 4: Up and down bars default formatting per chart style

This table lists line, fill, and effect formatting for up and down bars by style. The color listed as accent1-6 means that the first style uses accent1, the next uses accent2, up to the sixth uses accent6.

	Up Bars	Down Bars	Up & Down Bars			
			Themed			
Style	Fill Color	Fill Color	Fill	Effect	Line	Line Color
1	25% tint of dk1	85% tint of dk1	Subtle	None	Subtle	tx1
2	5% tint of dk1	95% tint of dk1	Subtle	None	Subtle	tx1
3-8	25% tint of accent1-6	25% shade of accent1-6	Subtle	None	Subtle	tx1
9	25% tint of dk1	85% tint of dk1	Subtle	Subtle	Subtle	tx1
10	5% tint of dk1	95% tint of dk1	Subtle	Subtle	Subtle	tx1
11-16	25% tint of accent1-6	25% shade of accent1-6	Subtle	Subtle	Subtle	tx1
17	25% tint of dk1	85% tint of dk1	Intense	Moderate	No ne	None
18	5% tint of dk1	95% tint of dk1	Intense	Moderate	None	None
19-24	25% tint of accent1-6	25% shade of accent1-6	Intense	Moderate	None	None
25	25% tint of dk1	85% tint of dk1	Intense	Intense	None	None
26	5% tint of dk1	95% tint of dk1	Intense	Intense	None	None

	Up Bars	Down Bars	Up & Down Bars			
27-32	25% tint of accent1-6	25% shade of accent1-6	Intense	Intense	None	None
33	lt1	85% tint of dk1	Subtle	None	Subtle	dk1
34	lt1	95% tint of dk1	Subtle	None	Subtle	dk1
35-40	lt1	25% shade of accent1-6	Subtle	None	Subtle	25% shade of accent1-6
41	25% tint of dk1	85% tint of dk1	Intense	Intense	None	None
42	lt1	dk1	Intense	Intense	None	None
43-48	25% tint of accent1-6	25% shade of accent1-6	Intense	Intense	None	None

Table 5: Default data point formatting per chart style

This table lists line, fill, and effect formatting for data points by style. Some of the formatting is a repeating pattern described in the next table (denoted by "Pattern"). Other use a fade pattern in which the first series is a certain% shade of the color listed and the last series is an certain% tint of the color listed. The intermediate colors are linearly interpolated by shade and tint to the listed color. The starting shade% and ending tint% are determined by the spreadsheet application, where shades are always darker than tints.

[*Note*: A suggested way to implement this percentage is by using the formula: shade/tint percent = -70 + 140 * (SeriesFormattingIndex / (HighestFormattingIndexOfAllSeriesOnAllGraphs + 1)). In this case the series formatting index is the `idx` attribute value for the series. Negative outputs are shades, and positive outputs are tints. *end note*]

[*Note*: The arrows across the top of the table indicate which columns apply for that row: i.e., for Fills, Lines, or Markers. So, for example, only the last two columns for Lines for Data Points apply. *end note*]

Fills for Data Points (2-D)	↓	↓		↓	↓			
Fills for Data Points (3-D)		↓	↓	↓	↓			
Lines for Data Points						↓	↓	
Markers for Data Points	↓	↓					↓	

Style	Effect Themed Effect	Fill Themed Fill	Fill Pattern	Fill Themed Fill	Line Themed Line	Line Color or Pattern	Line Width	Line Pattern
1	No Effect	Subtle	1	Subtle	No Line		3	1
2	No Effect	Subtle	2	Subtle	No Line		3	2
3	No Effect	Subtle	accent1 fade	Subtle	No Line		3	accent1 fade
4	No Effect	Subtle	accent2 fade	Subtle	No Line		3	accent2 fade
5	No Effect	Subtle	accent3 fade	Subtle	No Line		3	accent3 fade
6	No Effect	Subtle	accent4 fade	Subtle	No Line		3	accent4 fade
7	No Effect	Subtle	accent5 fade	Subtle	No Line		3	accent5 fade
8	No Effect	Subtle	accent6 fade	Subtle	No Line		3	accent6 fade
9	Subtle	Subtle	1	Subtle	Subtle	lt1	5	1
10	Subtle	Subtle	2	Subtle	Subtle	lt1	5	2
11	Subtle	Subtle	accent1 fade	Subtle	Subtle	lt1	5	accent1 fade
12	Subtle	Subtle	accent2 fade	Subtle	Subtle	lt1	5	accent2 fade
13	Subtle	Subtle	accent3 fade	Subtle	Subtle	lt1	5	accent3 fade
14	Subtle	Subtle	accent4 fade	Subtle	Subtle	lt1	5	accent4 fade
15	Subtle	Subtle	accent5 fade	Subtle	Subtle	lt1	5	accent5 fade
16	Subtle	Subtle	accent6	Subtle	Subtle	lt1	5	accent6

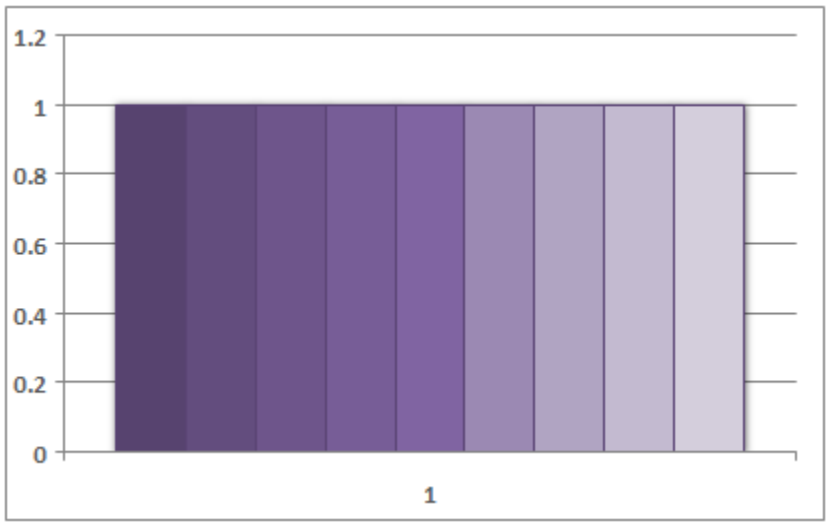
			fade					fade
17	Moderate	Intense	1	Subtle	No Line		5	1
18	Moderate	Intense	2	Intense	No Line		5	2
19	Moderate	Intense	accent1 fade	Subtle	No Line		5	accent1 fade
20	Moderate	Intense	accent2 fade	Subtle	No Line		5	accent2 fade
21	Moderate	Intense	accent3 fade	Subtle	No Line		5	accent3 fade
22	Moderate	Intense	accent4 fade	Subtle	No Line		5	accent4 fade
23	Moderate	Intense	accent5 fade	Subtle	No Line		5	accent5 fade
24	Moderate	Intense	accent6 fade	Subtle	No Line		5	accent6 fade
25	Intense	Intense	1	Subtle	No Line		7	1
26	Intense	Intense	2	Intense	No Line		7	2
27	Intense	Intense	accent1 fade	Subtle	No Line		7	accent1 fade
28	Intense	Intense	accent2 fade	Subtle	No Line		7	accent2 fade
29	Intense	Intense	accent3 fade	Subtle	No Line		7	accent3 fade
30	Intense	Intense	accent4 fade	Subtle	No Line		7	accent4 fade
31	Intense	Intense	accent5 fade	Subtle	No Line		7	accent5 fade
32	Intense	Intense	accent6 fade	Subtle	No Line		7	accent6 fade
33	No Effect	Subtle	1	Subtle	Subtle	50% shade	5	1

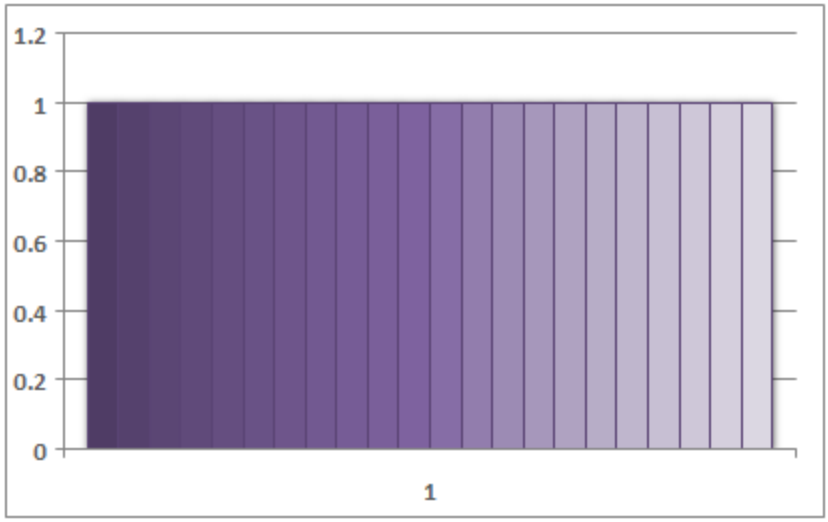
						of dk1		
34	No Effect	Subtle	2	Subtle	Subtle	Pattern 3	5	2
35	No Effect	Subtle	accent1 fade	Subtle	Subtle	50% shade of accent1	5	accent1 fade
36	No Effect	Subtle	accent2 fade	Subtle	Subtle	50% shade of accent2	5	accent2 fade
37	No Effect	Subtle	accent3 fade	Subtle	Subtle	50% shade of accent3	5	accent3 fade
38	No Effect	Subtle	accent4 fade	Subtle	Subtle	50% shade of accent4	5	accent4 fade
39	No Effect	Subtle	accent5 fade	Subtle	Subtle	50% shade of accent5	5	accent5 fade
40	No Effect	Subtle	accent6 fade	Subtle	Subtle	50% shade of accent6	5	accent6 fade
41	Intense	Intense	4	Subtle	No Line		5	4
42	Intense	Intense	2	Intense	No Line		5	2
43	Intense	Intense	accent1 fade	Subtle	No Line		5	accent1 fade
44	Intense	Intense	accent2 fade	Subtle	No Line		5	accent2 fade
45	Intense	Intense	accent3 fade	Subtle	No Line		5	accent3 fade
46	Intense	Intense	accent4 fade	Subtle	No Line		5	accent4 fade
47	Intense	Intense	accent5 fade	Subtle	No Line		5	accent5 fade
48	Intense	Intense	accent6 fade	Subtle	No Line		5	accent6 fade

Table 6: Default data point formatting per data point

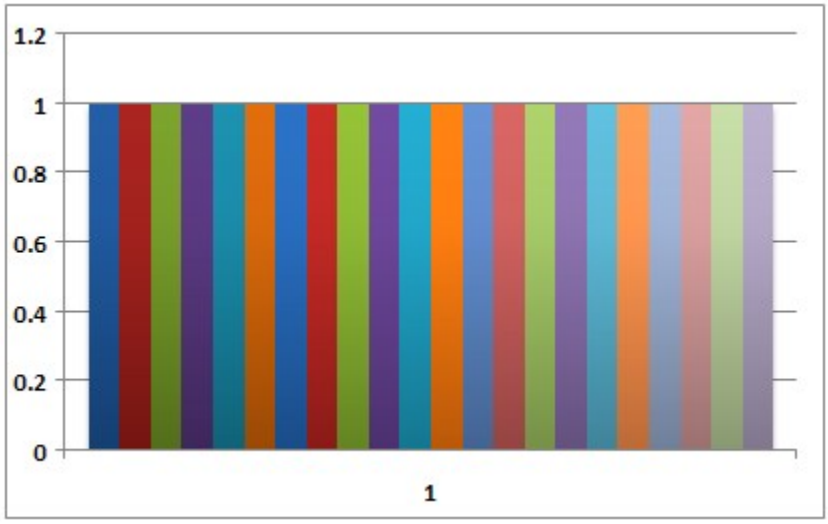
This table contains the formatting patterns used for each data point when there are is one series, or each series when there is just one series. The patterns in this table can repeat, see the pictures below the table for an illustration.

Pattern	Data Point 1	Data Point 2	Data Point 3	Data Point 4	Data Point 5	Data Point 6
1	88.5% tint of dk1	55% tint of dk1	78% tint of dk1	92.5% tint of dk1	70% tint of dk1	30% tint of dk1
2	accent1	accent2	accent3	accent4	Accent5	accent6
3	50% shade of accent1	50% shade of accent2	50% shade of accent3	50% shade of accent4	50% shade of accent5	50% shade of accent6
4	5% tint of dk1	55% tint of dk1	78% tint of dk1	15% tint of dk1	70% tint of dk1	30% tint of dk1





Above are two charts showing the same monochromatic pattern, just with a different number of series. [*Note*: These charts just have tint and shade adjusted for each series beginning and ending at the same values. *end note*]



Above is a pattern that has 6 distinct accent colors. The colors repeat every 6 data points, but the tint/shade is changed for each set.

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 1.
- This simple type has a maximum value of less than or equal to 48.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Style) is located in §A.5.1. *end note*]

21.2.3.47 ST_TickLblPos (Tick Label Position)

This simple type specifies the possible positions for tick labels.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
high (High)	Specifies the axis labels shall be at the high end of the perpendicular axis.
low (Low)	Specifies the axis labels shall be at the low end of the perpendicular axis.
nextTo (Next To)	Specifies the axis labels shall be next to the axis.
none (None)	Specifies the axis labels are not drawn.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TickLblPos) is located in §A.5.1. *end note*]

21.2.3.48 ST_TickMark (Tick Mark)

This simple type specifies the possible positions for tick marks.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
cross (Cross)	Specifies the tick marks shall cross the axis.
in (Inside)	Specifies the tick marks shall be inside the plot area.
none (None)	Specifies there shall be no tick marks.
out (Outside)	Specifies the tick marks shall be outside the plot area.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TickMark) is located in §A.5.1. *end note*]

21.2.3.49 ST_TimeUnit (Time Unit)

This simple type specifies a unit of time.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
days (Days)	Specifies the chart data shall be shown in days.

Enumeration Value	Description
months (Months)	Specifies the chart data shall be shown in months.
years (Years)	Specifies the chart data shall be shown in years.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TimeUnit) is located in §A.5.1. *end note*]

21.2.3.50 ST_TrendlineType (Trendline Type)

This simple type specifies all styles of trendline which are available for series in a chart.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
exp (Exponential)	Specifies the trendline shall be an exponential curve in the form $y = ab^x$.
linear (Linear)	Specifies the trendline shall be a line in the form $y = mx + b$.
log (Logarithmic)	Specifies the trendline shall be a logarithmic curve in the form $y = a \log x + b$, where log is the natural logarithm.
movingAvg (Moving Average)	Specifies the trendline shall be a moving average of period Period.
poly (Polynomial)	Specifies the trendline shall be a polynomial curve of order Order in the form $y = ax^6 + bx^5 + cx^4 + dx^3 + ex^2 + fx + g$.
power (Power)	Specifies the trendline shall be a power curve in the form $y = ax^b$.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TrendlineType) is located in §A.5.1. *end note*]

21.2.3.51 ST_DepthPercentWithSymbol (Depth Percent with Symbol)

This simple type specifies that its contents contain a percentage between 20% and 2000%.

The simple type's contents shall match the following regular expression pattern:
0*(([2-9][0-9])|([1-9][0-9][0-9])|(1[0-9][0-9][0-9])|2000)%.

21.2.3.52 ST_HPercentWithSymbol (Height Percent with Symbol)

This simple type specifies that its contents contain a percentage between 5% and 500%.

The simple type's contents shall match the following regular expression pattern:

0*(([5-9])|([1-9][0-9])|([1-4][0-9][0-9])|500)%.

21.2.3.53 ST_GapAmountPercent (Gap Amount Percentage)

This simple type specifies that its contents contain a percentage between 0% and 500%.

The simple type's contents shall match the following regular expression pattern:

0*(([0-9])|([1-9][0-9])|([1-4][0-9][0-9])|500)%.

21.2.3.54 ST_SecondPieSizePercent (Second Pie Size Percentage)

This simple type specifies that its contents contain a percentage between 5% and 200%.

The simple type's contents shall match the following regular expression pattern:

0* (([5-9])|([1-9][0-9])|(1[0-9][0-9])|200)%.

21.2.3.55 ST_HoleSizePercent (Hole Size Percentage)

This simple type specifies that its contents contain a percentage between 1% and 90%.

The simple type's contents shall match the following regular expression pattern: 0*([1-9]|([1-8][0-9])|90)%.

21.2.3.56 ST_LblOffsetPercent (Label Offset Percentage)

This simple type specifies that its contents contain a percentage between 0% and 1000%.

The simple type's contents shall match the following regular expression pattern:

0*(([0-9])|([1-9][0-9])|([1-9][0-9][0-9])|1000)%.

21.2.3.57 ST_OverlapPercent (Overlap Percentage)

This simple type specifies that its contents contain a percentage between -100% and 100%.

The simple type's contents shall match the following regular expression pattern:

(-?0*(([0-9])|([1-9][0-9])|100))%.

21.2.3.58 ST_BubbleScalePercent (Bubble Scale Percentage)

This simple type specifies that its contents contain a percentage between 0% and 300%.

The simple type's contents shall match the following regular expression pattern:

0*(([0-9])|([1-9][0-9])|([1-2][0-9][0-9])|300)%.

21.2.3.59 ST_Thickness (Thickness Percentage)

This simple type specifies that its contents contain a percentage.

This simple type is a union of the following types:

- ST_ThicknessPercent simple type (§21.2.3.60).

21.2.3.60 ST_ThicknessPercent (Thickness Percentage)

This simple type specifies that its contents contain a percentage.

The simple type's contents shall match the following regular expression pattern: ([0-9]+)%.

21.3 DrawingML - Chart Drawings

Within a chart, it is sometimes necessary to include DrawingML elements (shapes or pictures) which should be a child object within the parent chart. This relationship allows those elements to optionally be resized with the chart, automatically moved with the chart, etc.

The Chart Drawing namespace acts in this capacity, specifying all information necessary to anchor and display DrawingML objects within a chart.

21.3.1 Table of Contents

This subclause is informative.

End of informative text.

21.3.2 Elements

The following element define the contents of the ChartDrawing namespace:

21.3.2.1 absSizeAnchor (Absolute Anchor Shape Size)

This element specifies that the shape described here to reside within a chart should be sized based on relative anchor points. This is achieved via two elements. The from element specifies the top left corner of the shape bounding box in a RTL(right-to-left) implementation. The ext element then specifies the bottom right corner of the shape bounding box in a RTL(right-to-left) implementation and thus the size of the shape.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AbsSizeAnchor) is located in §A.5.2. *end note*]

21.3.2.2 blipFill (Picture Fill)

This element specifies the kind of picture fill that the picture object has. Because a picture has a picture fill already by default, it is possible to have two fills specified for a picture object. An example of this is shown below.

[*Example*: Consider the picture below that has a blip fill applied to it. The image used to fill this picture object has transparent pixels instead of white pixels.

```
<pic:pic>
  …
  <pic:blipFill>
    <a:blip r:embed="rId2"/>
    <a:stretch>
      <a:fillRect/>
    </a:stretch>
  </pic:blipFill>
  …
</pic:pic>
```

The above picture object is shown as an example of this fill kind. *end example*]

[*Example*: Consider now the same picture object but with an additional gradient fill applied within the shape properties portion of the picture.

```
<pic:pic>
  …
  <pic:blipFill>
    <a:blip r:embed="rId2"/>
    <a:stretch>
      <a:fillRect/>
    </a:stretch>
  </pic:blipFill>
  <pic:spPr>
    <a:gradFill>
      <a:gsLst>
        <a:gs pos="0">
          <a:schemeClr val="tx2">
            <a:shade val="50000"/>
          </a:schemeClr>
        </a:gs>
        <a:gs pos="39999">
          <a:schemeClr val="tx2">
            <a:tint val="20000"/>
          </a:schemeClr>
        </a:gs>
        <a:gs pos="70000">
          <a:srgbClr val="C4D6EB"/>
        </a:gs>
        <a:gs pos="100000">
          <a:schemeClr val="bg1"/>
```



```
        </a:gs>
      </a:gsLst>
    </a:gradFill>
  </pic:spPr>
  …
</pic:pic>
```

The above picture object is shown as an example of this double fill kind. *End example*]

Attributes	Description
dpi (DPI Setting) Namespace: http://purl.oclc.org/ooxml/drawingml/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: http://purl.oclc.org/ooxml/drawingml/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.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BlipFillProperties) is located in §A.4.1. *end note*]

21.3.2.3 cNvCxnSpPr (Non-Visual Connection Shape Drawing Properties)

This element specifies the non-visual drawing properties for a connector shape. These non-visual properties are properties that the generating application would utilize when rendering the parent chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualConnectorProperties) is located in §A.4.1. *end note*]

21.3.2.4 cNvGraphicFramePr (Non-Visual Graphic Frame Drawing Properties)

This element specifies the non-visual drawing properties for a graphic frame. These non-visual properties are properties that the generating application would utilize when rendering the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualGraphicFrameProperties) is located in §A.4.1. *end note*]

21.3.2.5 cNvGrpSpPr (Non-Visual Group Shape Drawing Properties)

This element specifies the non-visual drawing properties for a group shape. These non-visual properties are properties that the generating application would utilize when rendering the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualGroupDrawingShapeProps) is located in §A.4.1. *end note*]

21.3.2.6 cNvPicPr (Non-Visual Picture Drawing Properties)

This element specifies the non-visual properties for the picture canvas. These properties are to be used by the generating application to determine how certain properties are to be changed for the picture object in question.

[*Example*: Consider the following DrawingML.

```
<pic:pic>
  …
  <pic:nvPicPr>
    <pic:cNvPr id="4" name="Lilly_by_Lisher.jpg"/>
    <pic:cNvPicPr>
      <a:picLocks noChangeAspect="1"/>
    </pic:cNvPicPr>
    <pic:nvPr/>
  </pic:nvPicPr>
  …
</pic:pic>
```

End example]

Attributes	Description
preferRelativeResize (Relative Resize	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

Attributes	Description
Preferred) Namespace: http://purl.oclc.org/ooxml/drawingml/main	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.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualPictureProperties) is located in §A.4.1. *end note*]

21.3.2.7 cNvPr (Non-Visual Drawing Properties)

This element specifies non-visual canvas properties. This allows for additional information that does not affect the appearance of the picture to be stored.

[*Example*: Consider the following ChartDrawingML.

```
<cdr:pic>
  …
  <cdr:nvPicPr>
    <cdr:cNvPr id="4" name="Lilly_by_Lisher.jpg"/>
  </cdr:nvPicPr>
  …
</cdr:pic>
```

end example]

Attributes	Description
descr (Alternative Text for Object) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies alternative text for the current DrawingML object, for use by assistive technologies or applications which 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 which can be used in place of the actual

Attributes	Description
	DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
hidden (Hidden) Namespace: http://purl.oclc.org/ooxml/drawingml/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 which 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: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.21).
name (Name) Namespace: http://purl.oclc.org/ooxml/drawingml/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" >`

Attributes	Description
	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: http://purl.oclc.org/ooxml/drawingml/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.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualDrawingProps) is located in §A.4.1. *end note*]

21.3.2.8 cNvSpPr (Non-Visual Shape Drawing Properties)

This element specifies the non-visual drawing properties for a shape. These properties are to be used by the generating application to determine how the shape should be dealt with.

[*Example*: Consider the shape that has a shape lock applied to it.

```
<cdr:sp>
  <cdr:nvSpPr>
    <cdr:cNvPr id="2" name="Rectangle 1"/>
    <cdr:cNvSpPr>
      <a:spLocks noGrp="1"/>
    </cdr:cNvSpPr>
  </cdr:nvSpPr>
  …
</cdr:sp>
```

This shape lock is stored within the non-visual drawing properties for this shape. *end example*]

Attributes	Description
txBox (Text Box) Namespace:	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.

Attributes	Description
http://purl.oclc.org/ooxml/drawingml/main	[*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.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualDrawingShapeProps) is located in §A.4.1. *end note*]

21.3.2.9 cxnSp (Connection Shape)

This element specifies a connection shape that is used to connect two sp elements. Once a connection is specified using a cxnSp, it is left to the generating application to determine the exact path the connector takes. That is the connector routing algorithm is left up to the generating application as the desired path might be different depending on the specific needs of the application.

[*Example*:

end example]

[*Example*: Consider the following connector shape that connects two regular shapes.

```
<cdrcdr:grpSp>
  …
  <cdr:sp>
    <cdr:nvSpPr>
      <cdr:cNvPr id="1" name="Rectangle 1"/>
      <cdr:cNvSpPr/>
      <cdr:nvPr/>
    </cdr:nvSpPr>
    …
  </cdr:sp>
  <cdr:sp>
    <cdr:nvSpPr>
      <cdr:cNvPr id="2" name="Rectangle 2"/>
      <cdr:cNvSpPr/>
      <cdr:nvPr/>
```

```
      </cdr:nvSpPr>
      …
    </cdr:sp>
    <cdr:cxnSp>
      <cdr:nvCxnSpPr>
        <cdr:cNvPr id="3" name="Elbow Connector 3"/>
        <cdr:cNvCxnSpPr>
          <a:stCxn id="1" idx="3"/>
          <a:endCxn id="2" idx="1"/>
        </cdr:cNvCxnSpPr>
        <cdr:nvPr/>
      </cdr:nvCxnSpPr>
      …
    </cdr:cxnSp>
  </cdr:grpSp>
```

end example]

Attributes	Description

fPublished (Publish to Server)	Specifies whether the shape shall be published with the worksheet when sent to the spreadsheet server. This is for use when interfacing with a document server.
	
	[*Example*: Consider the following shape that is not published with the worksheet when it is published back on the spreadsheet server.
	`<cdr:relSizeAnchor>`
	`…`
	`<cdr:sp fPublished="0">`
	`..`
	`</cdr:sp>`
	`…`
	`</cdr:relSizeAnchor>`
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

macro (Reference to Custom Function)	This element specifies the custom function associated with the chart. [*Example*: A macro script, add-in function, and so on. *end example*] The format of this string shall be application-defined, and should be ignored if not understood. [*Example*: `<… macro="DoWork()" >` *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_Connector) is located in §A.5.2. *end note*]

21.3.2.10 ext (Shape Extent)

This element describes the length and width properties for how far a drawing element should extend for.

Attributes	Description
cx (Extent Length) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.41).
cy (Extent Width) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositiveSize2D) is located in §A.4.1. *end note*]

21.3.2.11 from (Starting Anchor Point)

This element specifies the first anchor point for the drawing element. This is used to anchor the top and left sides of the shape within the chart. That is when the corresponding chart is adjusted, the shape is also adjusted.

[*Example*: Consider the following Chart Drawing content:

```
<cdr:relSizeAnchor>
  <cdr:from>
    <cdr:x>0.04583</cdr:x>
    <cdr:y>0.53125</cdr:y>
  </cdr:from>
  <cdr:to>
    <cdr:x>0.24583</cdr:x>
    <cdr:y>0.86458</cdr:y>
  </cdr:to>
  <cdr:sp macro="" textlink="">
  …
  </cdr:sp>
</cdr:relSizeAnchor>
```

The above example shows the first anchor point being specified via the from element. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Marker) is located in §A.5.2. *end note*]

21.3.2.12 graphicFrame (Graphic Frame)

This element specifies the existence of a graphics frame. This frame contains a graphic that was generated by an external source and needs a container in which to be displayed on the slide surface.

Attributes	Description

fPublished (Publish To Server)	Specifies whether the shape shall be published with the worksheet when sent to the spreadsheet server. This is for use when interfacing with a document server.
	
	[*Example*: Consider the following shape that is not published with the worksheet when it is published back on the spreadsheet server.
	`<cdr:relSizeAnchor>`
	`…`
	`<cdr:sp fPublished="0">`
	`…`
	`</cdr:sp>`
	`…`
	`</cdr:relSizeAnchor>`
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

macro (Reference to Custom Function)	This element specifies the custom function associated with the chart. [*Example*: A macro script, add-in function, and so on. *end example*]

Attributes	Description
	The format of this string shall be application-defined, and should be ignored if not understood. [*Example*: `<… macro="DoWork()" >` *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_GraphicFrame) is located in §A.5.2. *end note*]

21.3.2.13 grpSp (Group Shape)

This element specifies a group shape that represents many shapes grouped together. This shape is to be treated just as if it were a regular shape but instead of being described by a single geometry it is made up of all the shape geometries encompassed within it. Within a group shape each of the shapes that make up the group are specified just as they normally would. The idea behind grouping elements however is that a single transform can apply to many shapes at the same time.

[*Example*: Consider the following group shape.

```
<cdr:grpSp>
  <cdr:nvGrpSpPr>
    <cdr:cNvPr id="10" name="Group 9"/>
    <cdr:cNvGrpSpPr/>
    <cdr:nvPr/>
  </cdr:nvGrpSpPr>
  <cdr:grpSpPr>
    <a:xfrm>
      <a:off x="838200" y="990600"/>
      <a:ext cx="2426208" cy="978408"/>
      <a:chOff x="838200" y="990600"/>
      <a:chExt cx="2426208" cy="978408"/>
   </a:xfrm>
  </cdr:grpSpPr>
  <cdr:sp>
  …
  </cdr:sp>
  <cdr:sp>
```

```
    …
    </cdr:sp>
    <cdr:sp>
    …
    </cdr:sp>
  </cdr:grpSp>
```

In the above example we see three shapes specified within a single group. These three shapes have their position and sizes specified just as they normally would within the shape tree. The generating application should apply the transformation after the bounding box for the group shape has been calculated. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupShape) is located in §A.5.2. *end note*]

21.3.2.14 grpSpPr (Group Shape Properties)

This element specifies the properties that are to be common across all of the shapes within the corresponding group. If there are any conflicting properties within the group shape properties and the individual shape properties then the individual shape properties should take precedence.

Attributes	Description
bwMode (Black and White Mode) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupShapeProperties) is located in §A.4.1. *end note*]

21.3.2.15 nvCxnSpPr (Connector Non Visual Properties)

This element specifies all non-visual properties for a connection shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a connection shape. This allows for additional information that does not affect the appearance of the connection shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ConnectorNonVisual) is located in §A.5.2. *end note*]

21.3.2.16 nvGraphicFramePr (Non-Visual Graphic Frame Properties)

This element specifies all non-visual properties for a graphic frame. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a graphic frame. This allows for additional information that does not affect the appearance of the graphic frame to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GraphicFrameNonVisual) is located in §A.5.2. *end note*]

21.3.2.17 nvGrpSpPr (Non-Visual Group Shape Properties)

This element specifies all non-visual properties for a group shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a group shape. This allows for additional information that does not affect the appearance of the group shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupShapeNonVisual) is located in §A.5.2. *end note*]

21.3.2.18 nvPicPr (Non-Visual Picture Properties)

This element specifies the non visual properties for a picture. This allows for additional information that does not affect the appearance of the picture to be stored.

[*Example*: Consider the following DrawingML.

```
<pic:pic>
  …
  <pic:nvPicPr>
    …
  </pic:nvPicPr>
  …
</pic:pic>
```

End example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_PictureNonVisual) is located in §A.5.2. *end note*]

21.3.2.19 nvSpPr (Non-Visual Shape Properties)

This element specifies all non-visual properties for a shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a shape. This allows for additional information that does not affect the appearance of the shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeNonVisual) is located in §A.5.2. *end note*]

21.3.2.20 pic (Picture)

This element specifies the existence of a picture object within the document.

[*Example*: Consider the following ChartDrawingML that specifies the existence of a picture within a document. This picture can have non-visual properties, a picture fill as well as shape properties attached to it.

```
<cdr:pic>
  <cdr:nvPicPr>
    <cdr:cNvPr id="4" name="lake.JPG" descr="Picture of a Lake" />
    <cdr:cNvPicPr>
      <a:picLocks noChangeAspect="1"/>
    </cdr:cNvPicPr>
    <cdr:nvPr/>
  </cdr:nvPicPr>
  <cdr:blipFill>
  …
  </cdr:blipFill>
  <cdr:spPr>
  …
  </cdr:spPr>
</cdr:pic>
```

end example]

Attributes	Description

fPublished (Publish to Server)	Specifies whether the shape shall be published with the worksheet when sent to the spreadsheet server. This is for use when interfacing with a document server.
	
	[*Example*: Consider the following shape that is not published with the worksheet when it is published back on the spreadsheet server.
	`<cdr:relSizeAnchor>`
	`…`
	`<cdr:sp fPublished="0">`
	`…`
	`</cdr:sp>`
	`…`
	`</cdr:relSizeAnchor>`
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

macro (Reference to Custom Function)	This element specifies the custom function associated with the chart. [*Example*: A macro script, add-in function, and so on. *end example*] The format of this string shall be application-defined, and should be ignored if not

Attributes	Description
	understood. [*Example*: `<… macro="DoWork()" >` *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_Picture) is located in §A.5.2. *end note*]

21.3.2.21 relSizeAnchor (Relative Anchor Shape Size)

This element specifies that the shape described here to reside within a chart should be sized based on relative anchor points. This is achieved via two elements. The from element specifies the top left corner of the shape bounding box in a RTL(right-to-left) implementation. The to element then specifies the bottom right corner of the shape bounding box in a RTL(right-to-left) implementation and thus the size of the shape.

[*Note*: The W3C XML Schema definition of this element's content model (CT_RelSizeAnchor) is located in §A.5.2. *end note*]

21.3.2.22 sp (Shape)

This element specifies the existence of a single shape. A shape can either be a preset or a custom geometry, defined using the DrawingML framework. In addition to geometry, each shape can have both visual and non-visual properties attached. Text and corresponding styling information can also be attached to a shape. This shape is specified along with all other shapes within either the shape tree or group shape elements.

Attributes	Description

fLocksText (Lock Text)	Specifies whether to allow for the editing of text within this shape when the worksheet on which the shape resides has been protected as defined by SpreadsheetML. This allows for the specifying of locked or "protected" text on a per-shape basis within a spreadsheet document. If this attribute is not specified then a value of 0, or false is assumed.
	
	[*Example*: Consider the following shape that does not have locked text on.
	`<cdr:relSizeAnchor>`
	`…`
	`<cdr:sp fLocksText="0">`
	`…`
	`</cdr:sp>`
	`…`
	`</cdr:relSizeAnchor>`

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

fPublished (Publish to Server)	Specifies whether the shape shall be published with the worksheet when sent to the spreadsheet server. This is for use when interfacing with a document server.
	
	[*Example*: Consider the following shape that is not published with the worksheet when it is published back on the spreadsheet server.
	`<cdr:relSizeAnchor>`
	`…`
	`<cdr:sp fPublished="0">`
	`…`
	`</cdr:sp>`
	`…`
	`</cdr:relSizeAnchor>`
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

macro (Reference to Custom Function)	This element specifies the custom function associated with the chart. [*Example*: A macro script, add-in function, and so on. *end example*] The format of this string shall be application-defined, and should be ignored if not understood. [*Example*: `<… macro="DoWork()" >` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

textlink (Text Link)	Specifies whether the text contained within this shape is linked to a cell within the spreadsheet. That is the text within the shape has the value defined in the referenced spreadsheet cell.
	
	[*Example*: Consider the following shape with text linked to cell A1.
	`<cdr:relSizeAnchor>`
	`…`
	`<cdr:sp macro="" textlink="A1">`
	`…`
	`</cdr:sp>`
	`…`
	`</cdr:relSizeAnchor>`

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_Shape) is located in §A.5.2. *end note*]

21.3.2.23 spPr (Shape Properties)

This element specifies the visual shape properties that can be applied to a special shape such as a connector shape or picture. These are the same properties that are allowed to describe the visual properties of a shape but are used here to describe additional object-specific properties within a document. This allows for these shapes to have both the properties of a shape as well as specific properties that are unique to only them.

Attributes	Description
bwMode (Black and White Mode) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeProperties) is located in §A.4.1. *end note*]

21.3.2.24 style (Shape Style)

The element specifies the style that is applied to a shape and the corresponding references for each of the style components such as lines and fills.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeStyle) is located in §A.4.1. *end note*]

21.3.2.25 to (Ending Anchor Point)

This element specifies the second anchor point for the drawing element. This is used to anchor the bottom and right sides of the shape within the spreadsheet. That is when the corresponding chart is adjusted, the shape is also adjusted.

[*Example*: Consider the following ChartDrawingML

```
<cdr:relSizeAnchor>
  <cdr:from>
    <cdr:x>0.04583</cdr:x>
    <cdr:y>0.53125</cdr:y>
  </cdr:from>
  <cdr:to>
    <cdr:x>0.24583</cdr:x>
    <cdr:y>0.86458</cdr:y>
  </cdr:to>
  <cdr:sp macro="" textlink="">
  …
  </cdr:sp>
</cdr:relSizeAnchor>
```

The above example shows the second anchor point being specified via the to element. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Marker) is located in §A.5.2. *end note*]

21.3.2.26 txBody (Shape Text Body)

This element specifies the existence of text to be contained within the corresponding shape. All visible text and visible text related properties are contained within this element. There can be multiple paragraphs and within paragraphs multiple runs of text.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBody) is located in §A.4.1. *end note*]

21.3.2.27 x (Relative X Coordinate)

This element specifies the relative x coordinate that is used to define the percentage-based horizontal position for a shape within a chart drawing object. The coordinate boundaries are specified within the corresponding simple type listed below.

The possible values for this element are defined by the ST_MarkerCoordinate simple type (§21.3.3.1).

[*Note*: The W3C XML Schema definition of this element's content model (ST_MarkerCoordinate) is located in §A.5.2. *end note*]

21.3.2.28 xfrm (Graphic Frame Transform)

This element specifies a 2-D transform to be applied to a Graphic Frame.

Attributes	Description
flipH (Horizontal Flip) Namespace: http://purl.oclc.org/ooxml/drawingml/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: http://purl.oclc.org/ooxml/drawingml/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: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Transform2D) is located in §A.4.1. *end note*]

21.3.2.29 y (Relative Y Coordinate)

This element specifies the relative y coordinate that is used to define the percentage-based vertical position for a shape within a chart drawing object. The coordinate boundaries are specified within the corresponding simple type listed below.

The possible values for this element are defined by the ST_MarkerCoordinate simple type (§21.3.3.1).

[*Note*: The W3C XML Schema definition of this element's content model (ST_MarkerCoordinate) is located in §A.5.2. *end note*]

21.3.3 Simple Types

This is the complete list of simple types dedicated to DrawingML – Chart Drawings.

21.3.3.1 ST_MarkerCoordinate (Chart Marker Coordinate Value)

This simple type specifies the chart marker coordinate value. It is to be represented as a fractional position between 0.0 and 1.0 of the chart width or height with 0.0 being the left or top edge.

This simple type's contents are a restriction of the W3C XML Schema double datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 0.0.
- This simple type has a maximum value of less than or equal to 1.0.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_MarkerCoordinate) is located in §A.5.2. *end note*]

21.4 DrawingML - Diagrams

A DrawingML diagram allows the definition of diagrams using DrawingML objects and constructs. This namespace defines the contents of a DrawingML diagram.

21.4.1 Table of Contents

This subclause is informative.

End of informative text.

21.4.2 Diagram Definition

This section specifies the elements which define the layout and hierarchy of a diagram based on its constituent nodes and connections.

21.4.2.1 adj (Shape Adjust)

Shape adjust value. These can be used to modify the adjust handles supported on various auto shapes. It is only possible to set the initial value, not to modify it using constraints and rules.

[*Example*: Consider the following example of the adj element in a DrawingML diagram:

```
<adjLst>
    <adj idx="2" val=".35" />
</adjLst>
```

In this example we have a single adjust handle being modified by setting its value to `0.35`. *end example*]

Attributes	Description
idx (Adjust Handle Index)	Adjust value index. Different shapes support different adjust handles. The possible values for this attribute are defined by the ST_Index1 simple type (§21.4.7.39).
val (Value)	An absolute value. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Adj) is located in §A.5.3. *end note*]

21.4.2.2 adjLst (Shape Adjust List)

This element is simply a list of shape adjusts.

[*Example*: Consider the following example of the adjLst element in a DrawingML diagram:

```
<adjLst>
   <adj idx="1" val="1.35" />
    <adj idx="2" val=".35" />
</adjLst>
```

In this example we have a two adjust handle being modified for the containing shape. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_AdjLst) is located in §A.5.3. *end note*]

21.4.2.3 alg (Algorithm)

The algorithm used by the containing layout node. The algorithm defines the behavior of the layout node along with the behavior and layout of the nested layout nodes.

[*Example*: Consider the following example of alg being used in a DrawingML diagram:

```
<layoutNode name="arrow">
    <varLst/>
    <alg type="tx" />
    <shape type="upArrow">
          <adjLst>
                <adj idx="2" val=".35" />
          </adjLst>
    </shape>
    <presOf axis="desOrSelf" ptType="node" />
    <ruleLst/>
</layoutNode>
```

In this example, the `tx` algorithm is being used to layout text within the containing layout node. *end example*]

Attributes	Description
rev (Revision Number)	The revision number of an algorithm. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
type (Algorithm Type)	Specifies the algorithm type. The possible values for this attribute are defined by the ST_AlgorithmType simple type (§21.4.7.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Algorithm) is located in §A.5.3. *end note*]

21.4.2.4 cat (Category)

This element specifies a category in the user interface where this layout definition displays to the user.

[*Example*: Consider the following example of a cat in a DrawingML diagram:

```
<catLst>
    <cat type="relationship" pri="19000" />
</catLst>
```

In this example we define a single category called `relationship` which has a priority of `19000`. *end example*]

Attributes	Description
pri (Priority)	The priority within the category for this diagram determines the order in which it displays in the user interface. Lower numbers are displayed at the beginning of the list. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
type (Category Type)	Specifies the category type associated with the element. The possible values for this attribute are defined by the W3C XML Schema anyURI datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Category) is located in §A.5.3. *end note*]

21.4.2.5 catLst (Category List)

This element is simply a list of cat elements.

[*Example*: Consider the following example of a catLst element in a DrawingML diagram:

```
<catLst>
    <cat type="list" pri="18000" />
    <cat type="relationship" pri="19000" />
</catLst>
```

In this example we define two different categories which are to be displayed in the user interface. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Categories) is located in §A.5.3. *end note*]

21.4.2.6 choose (Choose Element)

The choose element wraps if/else blocks into a choose block.

[*Example*: Consider the following example of a choose element in a DrawingML diagram:

```
<choose name="Name1">
    <if name="Name2" func="var" arg="dir" op="equ" val="norm">
          <alg type="snake">
                <param type="grDir" val="tL"/>
                <param type="flowDir" val="row"/>
                <param type="contDir" val="sameDir"/>
                <param type="off" val="ctr"/>
          </alg>
    </if>
    <else name="Name3">
```

```
            <alg type="snake">
                  <param type="grDir" val="tR"/>
                  <param type="flowDir" val="row"/>
                  <param type="contDir" val="sameDir"/>
                  <param type="off" val="ctr"/>
            </alg>
      </else>
  </choose>
```

In this example, a choose element is used to define two different sets of parameters associated with a snake algorithm depending upon the direction in which the user wants the algorithm to flow (RTL or LTR*). end example*]

Attributes	Description

name (Name)	A unique name associated with the choose statement.
	
	[*Example*: Consider the following example of a choose element in a DrawingML diagram:
	
	`<choose name="Name1">`
	
	…
	
	`</choose>`
	
	In this example, the choose element is named `Name1`. *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_Choose) is located in §A.5.3. *end note*]

21.4.2.7 clrData (Color Transform Sample Data)

This element defines the sample data that is to be used in the user interface controls regarding displaying color transforms for a given diagram. This sample data predefines a data model to be combined with a layout definition in order to create a diagram which a color transform can be applied and displayed to the user as an example of the color transform.

[*Example*: Consider the following example of a clrData element in a DrawingML diagram:

```
<clrData>
    <dataModel>
          <ptLst>
```



```
                <pt modelId="0" type="doc"/>
                <pt modelId="1"/>
                <pt modelId="2"/>
                <pt modelId="3"/>
                <pt modelId="4"/>
                <pt modelId="5"/>
                <pt modelId="6"/>
            </ptLst>
            <cxnLst>
                <cxn modelId="7" srcId="0" destId="1" srcOrd="0" destOrd="0"/>
                <cxn modelId="8" srcId="0" destId="2" srcOrd="1" destOrd="0"/>
                <cxn modelId="9" srcId="0" destId="3" srcOrd="2" destOrd="0"/>
                <cxn modelId="10" srcId="0" destId="4" srcOrd="3" destOrd="0"/>
                <cxn modelId="11" srcId="0" destId="5" srcOrd="4" destOrd="0"/>
                <cxn modelId="12" srcId="0" destId="6" srcOrd="5" destOrd="0"/>
            </cxnLst>
            <bg/>
            <whole/>
        </dataModel>
    </clrData>
```

In this example we define 6 points which all connect back to a seventh document type point. *end example*]

Attributes	Description
useDef (Use Default)	If the value of this attribute is `true`, the data model defined in the clrData element is ignored and a default data model is used instead. 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_SampleData) is located in §A.5.3. *end note*]

21.4.2.8 constr (Constraint)

This element is used to specify size, position of nodes, text values, and layout dependencies between nodes in a layout definition.

[*Example*: Consider the following example of a constraint list which contains some example constraints being defined and applied to layout nodes in the layout definition:

```
    <constrLst>
        <constr type="w" for="ch" forName="node1" refType="w" refForName=""/>
```

```
    <constr type="h" for="ch" forName="node1" refType="w" refFor="ch"
refForName="node1" op="equ" fact="0.6"/>
    <constr type="w" for="ch" forName="transition1" refType="w" refFor="ch"
refForName="node1" op="equ" fact="0.1"/>
    <constr type="primFontSz" for="ch" forName="node1" refForName="" op="equ"
val="100"/>
</constrLst>
```

In this example we can see constraints being defined for the width and height along with the primary font size for a layout node referenced by node1. The width for a transition is also specified. *end example*]

Attributes	Description
fact (Factor)	Factor used in a reference constraint or a rule in order to modify a referenced value by the factor defined. [*Example*: Consider the following example of fact in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, the width for transition1 is being defined as one-tenth the width of node1. *end example*] The possible values for this attribute are defined by the W3C XML Schema double datatype.
for (For)	Specifies the axis of layout nodes to apply a constraint or rule to. [*Example*: Consider the following example of for in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, the for attribute is specifying that node1 is a child node to the current layout node. *end example*] The possible values for this attribute are defined by the ST_ConstraintRelationship simple type (§21.4.7.20).
forName (For Name)	Specifies the name of the layout node to apply a constraint or rule to. [*Example*: Consider the following example of forName in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, forName is specifying the layout node named transition1 for its reference. *end example*]

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.
op (Operator)	The operator constraint used to evaluate the condition. [*Example*: Consider the following example of op in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, op is specifying an equality defined between the two referencing values. *end example*] The possible values for this attribute are defined by the ST_BoolOperator simple type (§21.4.7.10).
ptType (Data Point Type)	Specifies the type of data point to select. The possible values for this attribute are defined by the ST_ElementType simple type (§21.4.7.27).
refFor (Reference For)	The for value of the referenced constraint. [*Example*: Consider the following example of refFor in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, refFor is specifying the reference constraint is a child of the current layout node. *end example*] The possible values for this attribute are defined by the ST_ConstraintRelationship simple type (§21.4.7.20).
refForName (Reference For Name)	The name of the layout node referenced by a reference constraint. [*Example*: Consider the following example of refForName in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, refForName is specifying the layout node named node1 for its reference. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

Attributes	Description
refPtType (Reference Point Type)	The point type used int he referenced constraint. The possible values for this attribute are defined by the ST_ElementType simple type (§21.4.7.27).
refType (Reference Type)	Specifies the type of a reference constraint. [*Example*: Consider the following example of refType in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, refType is specifying referencing the width attribute of `node1`. *end example*] The possible values for this attribute are defined by the ST_ConstraintType simple type (§21.4.7.21).
type (Constraint Type)	Specifies the constraint to apply to this layout node. [*Example*: Consider the following example of type in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, type is specifying the width attribute of `transition1`. *end example*] The possible values for this attribute are defined by the ST_ConstraintType simple type (§21.4.7.21).
val (Value)	Specifies an absolute value instead of reference another constraint. [*Example*: Consider the following example of forName in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" val="10"/>` In this example, val is specifying the absolute value of the width of `transition1`. *end example*] The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Constraint) is located in §A.5.3. *end note*]

21.4.2.9 constrLst (Constraint List)

This element is simply a list of constraints.

[*Example*: Consider the following example of a constraint list which contains some example constraints which are being defined and applied to layout nodes in the layout definition:

```
<constrLst>
    <constr type="w" for="ch" forName="node1" refType="w" refForName=""/>
    <constr type="h" for="ch" forName="node1" refType="w" refFor="ch"
refForName="node1" op="equ" fact="0.6"/>
    <constr type="w" for="ch" forName="transition1" refType="w" refFor="ch"
refForName="node1" op="equ" fact="0.1"/>
    <constr type="primFontSz" for="ch" forName="node1" refForName="" op="equ"
val="100"/>
</constrLst>
```

In this example we can see constraints being defined for the width and height along with the primary font size for a layout node referenced by `node1`. The width for a transition is also specified. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Constraints) is located in §A.5.3. *end note*]

21.4.2.10 dataModel (Data Model)

The data for this instance of the diagram. Either a sample data model, or the data the user has entered.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DataModel) is located in §A.5.3. *end note*]

21.4.2.11 desc (Description)

This element holds a description for a layout definition. The description can be used to describe the qualities associated with a particular layout definition.

Attributes	Description
lang (Language)	The natural language of the title or description of this layout definition. The possible values for this attribute are defined by the W3C XML Schema string datatype.
val (Value)	The string which is used as the description of the layout definition. 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_Description) is located in §A.5.3. *end note*]

21.4.2.12 else (Else)

This element is similar to an else statement in a programming language in that it wraps elements which are to be used when the if conditionals are not true.

[*Example*: Consider the following example of an else element in a DrawingML diagram within the context of a choose statement:

```
<choose name="Name1">
    <if name="Name2" func="var" arg="dir" op="equ" val="norm">
          <alg type="snake">
                <param type="grDir" val="tL"/>
                <param type="flowDir" val="row"/>
                <param type="contDir" val="sameDir"/>
                <param type="off" val="ctr"/>
          </alg>
    </if>
    <else name="Name3">
          <alg type="snake">
                <param type="grDir" val="tR"/>
                <param type="flowDir" val="row"/>
                <param type="contDir" val="sameDir"/>
                <param type="off" val="ctr"/>
          </alg>
    </else>
</choose>
```

In this example, a else element is used to define a set of parameters associated with the snake algorithm when the diagram is reversed. *end example*]

Attributes	Description

name (Name)	A unique name associated with the choose statement.
	
	[*Example*: Consider the following example of a else element in a DrawingML diagram:
	
	`<else name="Name1">`
	
	…
	
	`</else>`
	
	In this example, the else element is named `Name1`. *end example*].
	
	The possible values for this attribute are defined by the W3C XML Schema string

Attributes	Description
	datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Otherwise) is located in §A.5.3. *end note*]

21.4.2.13 extLst (Extension List)

This element specifies an extension list, within which all future extensions are defined within ext elements.

The extension list along with corresponding future extensions is used to extend the storage capabilities of the DrawingML framework. This allows for various new types of data to be stored natively within the existing diagram syntax.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OfficeArtExtensionList) is located in §A.4.1. *end note*]

21.4.2.14 forEach (For Each)

A looping structure, similar to a `for` loop in a programming language, which defines what data model points use this layout node.

[*Example*: Consider the following example of a forEach being used within a DrawingML diagram:

```
<forEach name="Name5" ref="" axis="ch" ptType="node">
    <layoutNode name="node1" styleLbl="" moveWith="">
          <alg type="sp"/>
          <shape
xmlns:r="http://purl.oclc.org/ooxml/officeDocument/relationships" r:blip="">
                <adjLst/>
          </shape>
          <constrLst/>
    </layoutNode>
</forEach>
```

In this example, the forEach element creates a layout node, referenced by the name `node1`, for every associated data model point in the diagram. In this particular instance the forEach creates the layout node for every child of the current point node. *end example*]

Attributes	Description
axis (Axis)	Specifies the axis on which to select data from the data model. [*Example*: `axis="ch"` selects children of the current point node and `axis="des"` selects all descendants. *end example*]

Attributes	Description
	The possible values for this attribute are defined by the ST_AxisTypes simple type (§21.4.7.7).

cnt (Count)	Specifies the count of items to use in a data set.
	
	[*Example*: Consider the following example of a forEach in a DrawingML diagram:
	
	`<forEach name="Name5" ref="" axis="ch" ptType="node" cnt="2">`
	
	…
	
	`</forEach>`
	
	In this example, up to two children are obtained through this forEach. *end example*]
	
	The possible values for this attribute are defined by the ST_UnsignedInts simple type (§21.4.7.63).

hideLastTrans (Hide Last Transition)	In algorithms that support transitions, this attribute specifies that the last transition is not rendered. This allows for diagrams that start and end with a node. The possible values for this attribute are defined by the ST_Booleans simple type (§21.4.7.9).
name (Name)	A unique identifier for the layout node. The possible values for this attribute are defined by the W3C XML Schema string datatype.

ptType (Data Point Type)	Specifies the type of data point to select.
	
	[*Example*: Consider the following example of a forEach in a DrawingML diagram:
	
	`<forEach name="Name5" ref="" axis="ch" ptType="node" cnt="2">`
	
	…
	
	`</forEach>`
	
	In this example, the forEach selects all node type points in the set. *end example*]
	
	The possible values for this attribute are defined by the ST_ElementTypes simple type (§21.4.7.28).

ref (Reference)	When used on a for-each element, causes the specified for-each element to be used instead. The possible values for this attribute are defined by the W3C XML Schema string datatype.

Attributes	Description
st (Start)	Specifies where to start in a data set. [*Example*: Consider the following example of a forEach in a DrawingML diagram: `<presOf axis="desOrSelf" ptType="node" st="2"/>` In this example, the second element in the set is the first point returned. *end example*] The possible values for this attribute are defined by the ST_Ints simple type (§21.4.7.40).
step (Step)	Specifies the step to use in a data set. A step with a value of 2 returns every other item in the set. The possible values for this attribute are defined by the ST_Ints simple type (§21.4.7.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ForEach) is located in §A.5.3. *end note*]

21.4.2.15 if (If)

Like an if statement in a programming language, wraps elements which are to be used under the conditions defined by its attributes.

[*Example*: Consider the following example of an if element in a DrawingML diagram within the context of a choose statement:

```
<choose name="Name1">
    <if name="Name2" func="var" arg="dir" op="equ" val="norm">
          <alg type="snake">
                <param type="grDir" val="tL"/>
                <param type="flowDir" val="row"/>
                <param type="contDir" val="sameDir"/>
                <param type="off" val="ctr"/>
          </alg>
    </if>
    <else name="Name3">
          <alg type="snake">
                <param type="grDir" val="tR"/>
                <param type="flowDir" val="row"/>
                <param type="contDir" val="sameDir"/>
                <param type="off" val="ctr"/>
          </alg>
    </else>
</choose>
```

In this example, a if element is used to define a set of parameters associated with the snake algorithm when the diagram is in the normal direction. *end example*]

Attributes	Description
arg (Argument)	Specifies the variable to use as part of the test in an if element. Ignored unless the function attribute is set to "var". The possible values for this attribute are defined by the ST_FunctionArgument simple type (§21.4.7.31).
axis (Axis)	Specifies the axis on which to select data from the data model. [*Example*: `axis="ch"` selects children of the current point node and `axis="des"` selects all descendants. *end example*] The possible values for this attribute are defined by the ST_AxisTypes simple type (§21.4.7.7).

cnt (Count)	Specifies the count of items to use in a data set.
	
	[*Example*: Consider the following example of a forEach in a DrawingML diagram:
	
	`<forEach name="Name5" ref="" axis="ch" ptType="node" cnt="2">`
	
	…
	
	`</forEach>`
	
	In this example, up to two children are obtained through this forEach. *end example*]
	
	The possible values for this attribute are defined by the ST_UnsignedInts simple type (§21.4.7.63).

func (Function)	The function used to evaluate the if condition.
	
	[*Example*: Consider the following example of `func` being used in DrawingML:
	
	`<if name="Name2" func="var" arg="dir" op="equ" val="norm">`
	…
	`</if>`
	
	In this example, `func` is set to `var`. *end example*]
	
	The possible values for this attribute are defined by the ST_FunctionType simple type (§21.4.7.33).

hideLastTrans (Hide Last Transition)	In algorithms that support transitions, this attribute specifies that the last transition is not rendered. This allows for diagrams that start and end with a node. The possible values for this attribute are defined by the ST_Booleans simple type

Attributes	Description
	(§21.4.7.9).

name (Name)	A unique identifier for the layout node.
	
	The function used to evaluate the if condition.
	
	[*Example*: Consider the following example of name being used in DrawingML:
	
	`<if name="Name2" func="var" arg="dir" op="equ" val="norm">`
	`…`
	`</if>`
	
	In this example, the name attribute is set to Name2. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

op (Operator)	The operator used to evaluate the condition.
	
	[*Example*: Consider the following example of op being used in DrawingML:
	
	`<if name="Name2" func="var" arg="dir" op="equ" val="norm">`
	`…`
	`</if>`
	
	In this example, op is being used to test the equality of the argument and value. *end example*]
	
	The possible values for this attribute are defined by the ST_FunctionOperator simple type (§21.4.7.32).

ptType (Data Point Type)	Specifies the type of data point to select.
	
	[*Example*: Consider the following example of a forEach in a DrawingML diagram:
	
	`<forEach name="Name5" ref="" axis="ch" ptType="node" cnt="2">`
	
	`…`
	
	`</forEach>`
	
	In this example, the forEach selects all node type points in the set. *end example*]
	
	The possible values for this attribute are defined by the ST_ElementTypes simple type (§21.4.7.28).

st (Start)	Specifies where to start in a data set. [*Example*: Consider the following example of a forEach in a DrawingML diagram:

Attributes	Description
	<presOf axis="desOrSelf" ptType="node" st="2"/> In this example, the second element in the set is the first point returned. *end example*] The possible values for this attribute are defined by the ST_Ints simple type (§21.4.7.40).
step (Step)	Specifies the step to use in a data set. A step with a value of 2 returns every other item in the set. The possible values for this attribute are defined by the ST_Ints simple type (§21.4.7.40).
val (Value)	An absolute value. The possible values for this attribute are defined by the ST_FunctionValue simple type (§21.4.7.34).

[*Note*: The W3C XML Schema definition of this element's content model (CT_When) is located in §A.5.3. *end note*]

21.4.2.16 layoutDef (Layout Definition)

This element is the root element for defining a layout definition. The layout definition is defined through a set of nested layout nodes. The layout definition is responsible for defining the look of a diagram.

Attributes	Description
defStyle (Default Style)	This attribute defines a reference to a default style which is to be applied to the diagram. The possible values for this attribute are defined by the W3C XML Schema string datatype.
minVer (Minimum Version)	Minimum product version that can support this layout definition. The possible values for this attribute are defined by the W3C XML Schema string datatype.
uniqueId (Unique Identifier)	The unique identifier for this layout definition. 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_DiagramDefinition) is located in §A.5.3. *end note*]

21.4.2.17 layoutDefHdr (Layout Definition Header)

This element is the header information representing the minimum knowledge needed by an application to preload information about a layout definition. This preloading allows for the actual load of the layout definition to occur at a later time which helps with any performance concerns an application might have.

[*Example*: Consider the following example of a layoutDefHdr within a DrawingML diagram:

```
<layoutDefHdr uniqueId="urn:layout/default">
    <title val="Basic Block List" />
    <desc val="" />
    <catLst>
        <cat type="list" pri="1000" />
    </catLst>
</layoutDefHdr>
```

In this example we define a title along with a category and prioritization for the diagram referenced by the uniqueId of urn:layout:default. *end example*]

Attributes	Description
defStyle (Default Style)	This attribute defines a reference to a default style which is to be applied to the diagram. The possible values for this attribute are defined by the W3C XML Schema string datatype.
minVer (Minimum Version)	Minimum product version that can support this Diagram Layout. The possible values for this attribute are defined by the W3C XML Schema string datatype.
resId (Resource Identifier)	Resource ID used internally. The possible values for this attribute are defined by the W3C XML Schema int datatype.
uniqueId (Unique Identifier)	The unique identifier for this layout definition. 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_DiagramDefinitionHeader) is located in §A.5.3. *end note*]

21.4.2.18 layoutDefHdrLst (Diagram Layout Header List)

This element is simply a list of layout definition headers. This list of headers is used internally as a way to group all of the layout definition headers together into a single structure.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DiagramDefinitionHeaderLst) is located in §A.5.3. *end note*]

21.4.2.19 layoutNode (Layout Node)

The layout node is the basic building block of diagrams. The layout node is responsible for defining how shapes are arranged in a diagram and how the data maps to a particular shape in a diagram.

[*Example*: Consider the following example of a basic layout node defined in a DrawingML diagram:

```
<layoutNode name="node">
    <varLst>
          <bulletEnabled val="1"/>
    </varLst>
    <presOf axis="desOrSelf" ptType="node"/>
    <alg type="tx"/>
    <shape type="rect"
xmlns:r="http://purl.oclc.org/ooxml/officeDocument/relationships" r:blip="">
          <adjLst/>
    </shape>
    <constrLst/>
    <ruleLst>
          <rule type="primFontSz" forName="" val="2" fact="NaN" max="NaN"/>
    </ruleLst>
</layoutNode>
```

In this example we define a layout node which holds text and is a rectangle. *end example*]

Attributes	Description
chOrder (Child Order)	Specifes the ordering of the child layout nodes for a given layout node. The possible values for this attribute are defined by the ST_ChildOrderType simple type (§21.4.7.15).
moveWith (Move With)	Reference to another layout node that this layout node moves with. The possible values for this attribute are defined by the W3C XML Schema string datatype.
name (Name)	A unique identifier for the layout node. The possible values for this attribute are defined by the W3C XML Schema string datatype.
styleLbl (Style Label)	Specify which formatting option from a style or color variation should be applied to this layout node. The possible values for this attribute are defined by the W3C XML Schema string

Attributes	Description
	datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_LayoutNode) is located in §A.5.3. *end note*]

21.4.2.20 param (Parameter)

The parameter element modifies the default behavior of an algorithm.

[*Example*: Consider the following example of a param being used in a DrawingML diagram:

```
<alg type="snake">
    <param type="grDir" val="tL"/>
    <param type="flowDir" val="row"/>
    <param type="contDir" val="sameDir"/>
    <param type="off" val="ctr"/>
</alg>
```

In this example we see the snake algorithm being utilized and four parameters being set which are associated with the snake algorithm and modify its behavior. *end example*]

Attributes	Description
type (Parameter Type)	Specifies the parameter which is being modified. The possible values for this attribute are defined by the ST_ParameterId simple type (§21.4.7.49).
val (Value)	Specifies the actual value to be given to the parameter type defined by the type attribute. The possible values for this attribute are defined by the ST_ParameterVal simple type (§21.4.7.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Parameter) is located in §A.5.3. *end note*]

21.4.2.21 presOf (Presentation Of)

This element specifies a particular data model point which is to be mapped to the containing layout node. This attribute is responsible for defining the mapping of data to the layout nodes in a diagram.

[*Example*: Consider the following example of presOf in use within a DrawingML diagram:

```
<presOf axis="desOrSelf" ptType="node"/>
```

In this example the presOf element is mapping to a particular data model point. *end example*]

Attributes	Description
axis (Axis)	Specifies the axis on which to select data from the data model. [*Example*: `axis="ch"` selects children of the current point node and `axis="des"` selects all descendants. *end example*] The possible values for this attribute are defined by the ST_AxisTypes simple type (§21.4.7.7).

cnt (Count)	Specifies the count of items to use in a data set.
	
	[*Example*: Consider the following example of a forEach in a DrawingML diagram:
	
	`<forEach name="Name5" ref="" axis="ch" ptType="node" cnt="2">`
	
	…
	
	`</forEach>`
	
	In this example, up to two children are obtained through this forEach. *end example*]
	
	The possible values for this attribute are defined by the ST_UnsignedInts simple type (§21.4.7.63).

hideLastTrans (Hide Last Transition)	In algorithms that support transitions, this attribute specifies that the last transition is not rendered. This allows for diagrams that start and end with a node. The possible values for this attribute are defined by the ST_Booleans simple type (§21.4.7.9).

ptType (Data Point Type)	Specifies the type of data point to select.
	
	[*Example*: Consider the following example of a forEach in a DrawingML diagram:
	
	`<forEach name="Name5" ref="" axis="ch" ptType="node" cnt="2">`
	
	…
	
	`</forEach>`
	
	In this example, the forEach selects all node type points in the set. *end example*]
	
	The possible values for this attribute are defined by the ST_ElementTypes simple type (§21.4.7.28).

st (Start)	Specifies where to start in a data set. [*Example*: Consider the following example of a forEach in a DrawingML diagram:

Attributes	Description
	<presOf axis="desOrSelf" ptType="node" st="2"/> In this example, the second element in the set is the first point returned. *end example*] The possible values for this attribute are defined by the ST_Ints simple type (§21.4.7.40).
step (Step)	Specifies the step to use in a data set. A step with a value of 2 returns every other item in the set. The possible values for this attribute are defined by the ST_Ints simple type (§21.4.7.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PresentationOf) is located in §A.5.3. *end note*]

21.4.2.22 relIds (Explicit Relationships to Diagram Parts)

This element specifies the relationship IDs used to explicitly reference each of the four constituent parts of a DrawingML diagram:

- Diagram Colors (cs attribute)
- Diagram Data (dm attribute)
- Diagram Layout Definition (lo attribute)
- Diagram Style (qs attribute)

Attributes	Description
cs (Explicit Relationship to Diagram Colors Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	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. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
dm (Explicit Relationship to Diagram Data Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/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 (§22.8.2.1).

Attributes	Description
lo (Explicit Relationship to Diagram Layout Definition Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/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 (§22.8.2.1).
qs (Explicit Relationship to Style Definition Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/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 (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RelIds) is located in §A.5.3. *end note*]

21.4.2.23 resizeHandles (Shape Resize Style)

This element defines the behavior when resizing shapes within a diagram. Because the size of the shape plays a large role in the overall layout of other nodes within the diagram, there are two ways resize can occur on a node.

Attributes	Description
val (Shape Resize Style Type)	Specifies the behavior for a shape when resizing shapes within a diagram. If this attribute is not specified, the default value shall be `rel`. The possible values for this attribute are defined by the ST_ResizeHandlesStr simple type (§21.4.7.54).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ResizeHandles) is located in §A.5.3. *end note*]

21.4.2.24 rule (Rule)

This element allows for a rule to be specified which changes the value of an existing constraint.

[*Example*: Consider the following example of a rule in a DrawingML diagram:

```
<ruleLst>
    <rule type="primFontSz" val="2" />
</ruleLst>
```

In this example a rule is being defined that shrinks the primary font size down to a lower limit of 2pt font when the text no longer fits correctly in the layout node. *end example*]

Attributes	Description
fact (Factor)	Factor used in a reference constraint or a rule in order to modify a referenced value by the factor defined. [*Example*: Consider the following example of fact in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, the width for `transition1` is being defined as one-tenth the width of `node1`. *end example*] The possible values for this attribute are defined by the W3C XML Schema double datatype.
for (For)	Specifies the axis of layout nodes to apply a constraint or rule to. [*Example*: Consider the following example of for in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, the for attribute is specifying that `node1` is a child node to the current layout node. *end example*] The possible values for this attribute are defined by the ST_ConstraintRelationship simple type (§21.4.7.20).
forName (For Name)	Specifies the name of the layout node to apply a constraint or rule to. [*Example*: Consider the following example of forName in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, forName is specifying the layout node named `transition1` for its reference. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

Attributes	Description
max (Max Value)	Sets the maximum value for a constraint so rules can no longer increase the constraint beyond that value. The possible values for this attribute are defined by the W3C XML Schema double datatype.
ptType (Data Point Type)	Specifies the type of data point to select. The possible values for this attribute are defined by the ST_ElementType simple type (§21.4.7.27).
type (Constraint Type)	Specifies the constraint to apply to this layout node. [*Example*: Consider the following example of type in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, type is specifying the width attribute of `transition1`. *end example*] The possible values for this attribute are defined by the ST_ConstraintType simple type (§21.4.7.21).
val (Value)	Specifies an absolute value instead of reference another constraint. [*Example*: Consider the following example of forName in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" val="10"/>` In this example, val is specifying the absolute value of the width of `transition1`. *end example*] The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumericRule) is located in §A.5.3. *end note*]

21.4.2.25 ruleLst (Rule List)

This element is simply a list of rules.

This element allows for a rule to be specified which changes the value of an existing constraint.

[*Example*: Consider the following example of a ruleLst in a DrawingML diagram:

```
<ruleLst>
    <rule type="primFontSz" val="2" />
```



```
</ruleLst>
```

In this example a single rule is being defined in the ruleLst that shrinks the primary font size down to a lower limit of 2pt font when the text no longer fits correctly in the layout node. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rules) is located in §A.5.3. *end note*]

21.4.2.26 sampData (Sample Data)

This element defines the sample data model which is used to pre-populate a diagram with placeholder data in order for the diagram to display itself in the user interface which shows all of the available diagrams to a user.

[*Example*: Consider the following example of a sampData element within a DrawingML diagram:

```
<sampData>
    <dataModel>
          <ptLst>
                <pt modelId="0" type="doc"/>
                <pt modelId="1">
                      <prSet phldr="1"/>
                </pt>
                <pt modelId="2">
                      <prSet phldr="1"/>
                </pt>
                <pt modelId="3">
                      <prSet phldr="1"/>
                </pt>
                <pt modelId="4">
                      <prSet phldr="1"/>
                </pt>
                <pt modelId="5">
                      <prSet phldr="1"/>
                </pt>
          </ptLst>
          <cxnLst>
                <cxn modelId="6" srcId="0" destId="1" srcOrd="0" destOrd="0"/>
                <cxn modelId="7" srcId="0" destId="2" srcOrd="1" destOrd="0"/>
                <cxn modelId="8" srcId="0" destId="3" srcOrd="2" destOrd="0"/>
                <cxn modelId="9" srcId="0" destId="4" srcOrd="3" destOrd="0"/>
                <cxn modelId="10" srcId="0" destId="5" srcOrd="4" destOrd="0"/>
          </cxnLst>
          <bg/>
          <whole/>
    </dataModel>
```

```
</sampData>
```

In this example we define the sample data to consist of five nodes all attached to a document point type node. When displayed, this diagram shows five shapes in the diagram. *end example*]

Attributes	Description
useDef (Use Default)	If the value of this attribute is `true`, the data model defined in the clrData element is ignored and a default data model is used instead. 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_SampleData) is located in §A.5.3. *end note*]

21.4.2.27 shape (Shape)

The shape displayed by the containing layout node. Not all layout nodes display shapes.

Attributes	Description
blip (Relationship to Image Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID of the explicit relationship to an image which 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 (§22.8.2.1).
blipPhldr (Image Placeholder)	Specifies whether to use an image placeholder or not. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
hideGeom (Hide Geometry)	When set to "true", hides the geometry of the shape. The text is still visible. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
lkTxEntry (Prevent Text Editing)	Prevents text editing on this shape. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rot (Rotation)	Rotates the shape by the specified number of degrees. The possible values for this attribute are defined by the W3C XML Schema double datatype.

Attributes	Description
type (Shape Type)	Specifies the type of shape. The possible values for this attribute are defined by the ST_LayoutShapeType simple type (§21.4.7.41).
zOrderOff (Z-Order Offset)	Offsets the shape from its default z-order stacking, which is based on the order the layout nodes appear in the XML. 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_Shape) is located in §A.5.3. *end note*]

21.4.2.28 style (Shape Style)

This element specifies the style information for a shape, as defined by its DrawingML child elements.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeStyle) is located in §A.4.1. *end note*]

21.4.2.29 styleData (Style Data)

This element defines the style data model which is used to pre-populate a diagram with placeholder data in order for the diagram to display itself in the user interface which shows a quick style applied to the diagram.

[*Example*: Consider the following example of a styleData being used within a DrawingML diagram:

```
<styleData>
    <dataModel>
          <ptLst>
                <pt modelId="0" type="doc"/>
                <pt modelId="1"/>
                <pt modelId="2"/>
          </ptLst>
          <cxnLst>
                <cxn modelId="3" srcId="0" destId="1" srcOrd="0" destOrd="0"/>
                <cxn modelId="4" srcId="0" destId="2" srcOrd="1" destOrd="0"/>
          </cxnLst>
          <bg/>
          <whole/>
    </dataModel>
</styleData>
```

In this example we define a data model which has only two nodes which are shown in the user interface when a layout definition is combined with this data model. *end example*]

Attributes	Description
useDef (Use Default)	If the value of this attribute is `true`, the data model defined in the clrData element is ignored and a default data model is used instead. 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_SampleData) is located in §A.5.3. *end note*]

21.4.2.30 title (Title)

Title of the Diagram Layout.

Attributes	Description
lang (Language)	Specifies the language of the title or description of this layout definition. The possible values for this attribute are defined by the W3C XML Schema string datatype.
val (Value)	Specifies the title or description of this layout definition. 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_Name) is located in §A.5.3. *end note*]

21.4.2.31 varLst (Variable List)

This element consists of a list of variables which interact with user interface components.

[*Example*: Consider the following example of a varLst in a DrawingML diagram:

```
<varLst>
    <chMax val="2" />
    <dir val="norm" />
    <resizeHandles val="exact" />
</varLst>
```

In this example we see different variables being defined which modify the behavior of user interface components directly. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_LayoutVariablePropertySet) is located in §A.5.3. *end note*]

21.4.3 Data

This section specifies the data that is to be contained within a diagram.

21.4.3.1 bg (Background Formatting)

This element defines formatting that can be applied to the background shape of the entire diagram. The background shape can hold formatting options just as a normal shape can hold within DrawingML.

[*Example*: Consider the following example of a bg in DiagramML:

```
<bg>
     <solidFill>
          <schemeClr val="tx1"/>
     </solidFill>
     <effectLst>
          <glow rad="152400">
               <schemeClr val="accent1">
                    <alpha val="75000"/>
               </schemeClr>
          </glow>
     </effectLst>
</bg>
```

In this example we see a solid fill applied to the background of the diagram along with a glow. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_BackgroundFormatting) is located in §A.4.1. *end note*]

21.4.3.2 cxn (Connection)

This element defines a connection between two points. A connection defines a relationship between two points in a diagram.

[*Example*: Consider the following example of a cxn in DiagramML:

```
<cxnLst>
    <cxn modelId="7" srcId="0" destId="1" srcOrd="0" destOrd="0"/>
    <cxn modelId="8" srcId="0" destId="2" srcOrd="1" destOrd="0"/>
    <cxn modelId="9" srcId="0" destId="3" srcOrd="2" destOrd="0"/>
    <cxn modelId="10" srcId="0" destId="4" srcOrd="3" destOrd="0"/>
    <cxn modelId="11" srcId="0" destId="5" srcOrd="4" destOrd="0"/>
    <cxn modelId="12" srcId="0" destId="6" srcOrd="5" destOrd="0"/>
</cxnLst>
```

In this example we see 6 cxn elements defined within a cxnLst element (§21.4.3.3). In this example, a relationship is being defined between point 0 and every other point in the diagram. *end example*]

Attributes	Description
destId (Destination Identifier)	The model identifier of the destination point for a connection. [*Example*: Consider the following example cxn within DiagramML: `<cxn modelId="10" srcId="0" destId="4" srcOrd="3" destOrd="0"/>` In this example we see the destination identifier referencing a point who's model identifier is 4. *end example*] The possible values for this attribute are defined by the ST_ModelId simple type (§21.4.7.43).
destOrd (Destination Position)	The relative position of the destination point among it's siblings. [*Example*: Consider the following example cxn within DiagramML: `<cxn modelId="10" srcId="0" destId="4" srcOrd="3" destOrd="0"/>` In this example we see the destination position is 0. This means that it is ranked first among its siblings if there are sibling points present. *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
modelId (Model Identifier)	The unique identifier associated with this cxn. [*Example*: Consider the following example cxn within DiagramML: `<cxn modelId="10" srcId="0" destId="4" srcOrd="3" destOrd="0"/>` In this example we see the model identifier is 10. *end example*] The possible values for this attribute are defined by the ST_ModelId simple type (§21.4.7.43).
parTransId (Parent Transition Identifier)	The model identifier of the point representing the parent transition. [*Example*: One example of a parent transition can be thought of as the shape connecting two points, such as an arrow in the diagram. *end example*] The unique identifier associated with this cxn. [*Example*: Consider the following example cxn within DiagramML: `<cxn modelId="10" srcId="0" destId="4" srcOrd="3" destOrd="0" parTransId="9" sibTransId="5"/>` In this example we see the parent transition identifier is referencing a point who's model identifier is 9. *end example*]

Attributes	Description
	The possible values for this attribute are defined by the ST_ModelId simple type (§21.4.7.43).
presId (Presentation Identifier)	The unique identifier of the layout associated to the cxn (only the active presentation (layout) is saved so all the presId's in the file should be the same). [*Example*: Consider the following example cxn within DiagramML: `<cxn modelId="10" type="presParOf" srcId="0" destId="4" srcOrd="3" destOrd="0" presId="urn:sampleLayouts/layout1"/>` In this example we see the presentation identifier is `urn:sampleLayouts/layout1`. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
sibTransId (Sibling Transition Identifier)	The model identifier of the point representing the sibling transition. [*Example*: An example of a sibling transition can be thought of as the shape connecting two points, such as an arrow in the diagram. *end example*] [*Example*: Consider the following example cxn within DiagramML: `<cxn modelId="10" srcId="0" destId="4" srcOrd="3" destOrd="0" parTransId="9" sibTransId="5"/>` In this example we see the sibling transition identifier is referencing a point who's model identifier is `5`. *end example*] The possible values for this attribute are defined by the ST_ModelId simple type (§21.4.7.43).
srcId (Source Identifier)	The model identifier of the source point for a connection. [*Example*: Consider the following example cxn within DiagramML: `<cxn modelId="10" srcId="0" destId="4" srcOrd="3" destOrd="0"/>` In this example we see the souce identifier referencing a point who's model identifier is `0`. *end example*] The possible values for this attribute are defined by the ST_ModelId simple type (§21.4.7.43).
srcOrd (Source Position)	The relative position of the source point among it's siblings. [*Example*: Consider the following example cxn within DiagramML:

Attributes	Description
	<cxn modelId="10" srcId="0" destId="4" srcOrd="3" destOrd="0"/> In this example we see the source position is 3. This means that it is ranked third among its siblings. *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
type (Point Type)	The type of point, which corresponds to a connection in this case. [*Example*: Consider the following example cxn within DiagramML: <cxn modelId="10" type="presParOf" srcId="0" destId="4" srcOrd="3" destOrd="0" presId="urn:sampleLayouts/layout1"/> In this example we see the point type is defined as presParOf. *end example*] The possible values for this attribute are defined by the ST_CxnType simple type (§21.4.7.23).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Cxn) is located in §A.5.3. *end note*]

21.4.3.3 cxnLst (Connection List)

This element defines a group of connections. There can be a connection list defined for any data model which holds all of the connections between points defined in the diagram.

[*Example*: Consider the following example of a cxnLst in DiagramML:

```
<cxnLst>
    <cxn modelId="7" srcId="0" destId="1" srcOrd="0" destOrd="0"/>
    <cxn modelId="8" srcId="0" destId="2" srcOrd="1" destOrd="0"/>
    <cxn modelId="9" srcId="0" destId="3" srcOrd="2" destOrd="0"/>
    <cxn modelId="10" srcId="0" destId="4" srcOrd="3" destOrd="0"/>
    <cxn modelId="11" srcId="0" destId="5" srcOrd="4" destOrd="0"/>
    <cxn modelId="12" srcId="0" destId="6" srcOrd="5" destOrd="0"/>
</cxnLst>
```

In this example we see 6 cxn elements (§21.4.3.2) defined within a cxnLst element. In this example, a relationship is being defined between point 0 and every other point in the diagram. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_CxnList) is located in §A.5.3. *end note*]

21.4.3.4 prSet (Property Set)

This element holds properties and customizations which are used throughout certain elements in DiagramML. The properties can be grouped into the following general categories:

- Presentation Properties - presLayoutVars, style, presAssocId, presName, presStyleLbl, presStyleIdx, presStyleCnt
- Document Properties - loTypeId, loCatId, qsTypeId, qaCatId, csTypeId, coherent3DOff
- Semantic Element Properties - phldrT, phldr
- Customization Properties - custAng, custFlipVert, custFlipHor, custSzX, custSzY, custScaleX, custScaleY, custT, custLinFactX, custLinFactY, custLinFactNeighborX, custLinFactNeighborY, custRadScaleRad, custRadScaleInc

[*Example*: Consider the basic example of prSet in use in DrawingML on a document point type:

```
<prSet loTypeId="urn:microsoft.com/office/officeart/2005/8/layout/default"
loCatId="list"
qsTypeId="urn:microsoft.com/office/officeart/2005/8/quickstyle/3d1" qsCatId="3D"
csTypeId="urn:microsoft.com/office/officeart/2005/8/colors/colorful2"
csCatId="colorful" phldr="1"/>
```

In this example we define the layout identifier, the category of the layout, the quick style identifier, the quick style category, along with the color style and color style category. *end example*]

Attributes	Description
coherent3DOff (Coherent 3D Behavior)	Enables or disables the Coherent 3D behavior for styles that specify this property. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
csCatId (Color Transform Category)	This attribute specifies the identifier of the current color transform category. The possible values for this attribute are defined by the W3C XML Schema string datatype.
csTypeId (Color Transform Type Identifier)	This attribute specifies the identifier of the currently applied color transform. The possible values for this attribute are defined by the W3C XML Schema string datatype.
custAng (Custom Rotation)	Specifies the amount that rotation is customized by. The possible values for this attribute are defined by the W3C XML Schema int datatype.
custFlipHor (Custom Horizontal Flip)	Specifies if there is a custom horizontal flip applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
custFlipVert	Specifies if there is a custom vertical flip applied.

Attributes	Description
(Custom Vertical Flip)	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
custLinFactNeighborX (Neighbor Offset Width)	Specifies the percentage of the neighbor's width used for offsetting shape. The possible values for this attribute are defined by ST_PrSetCustVal simple type (§21.4.7.66).
custLinFactNeighborY (Neighbor Offset Height)	Specifies the percentage of the neighbor's height used for offsetting shape. The possible values for this attribute are defined by ST_PrSetCustVal simple type (§21.4.7.66).
custLinFactX (Custom Factor Width)	Specifies the percentage of the current shape width used for offsetting the shape. The possible values for this attribute are defined by ST_PrSetCustVal simple type (§21.4.7.66).
custLinFactY (Custom Factor Height)	Specifies the percentage of the current shape height used for offsetting the shape. The possible values for this attribute are defined by ST_PrSetCustVal simple type (§21.4.7.66).
custRadScaleInc (Include Angle Scale)	Specifies the amount that the include angle has been scaled by. The possible values for this attribute are defined by ST_PrSetCustVal simple type (§21.4.7.66).
custRadScaleRad (Radius Scale)	Specifies how much the radius has been scaled. The possible values for this attribute are defined by ST_PrSetCustVal simple type (§21.4.7.66).
custScaleX (Width Scale)	Specifies the amount that the width has been scaled by. The possible values for this attribute are defined by ST_PrSetCustVal simple type (§21.4.7.66).
custScaleY (Height Scale)	Specifies the amount that the height has been scaled by. The possible values for this attribute are defined by ST_PrSetCustVal simple type (§21.4.7.66).
custSzX (Fixed Width Override)	Specifies a fixed width override for a shape. The possible values for this attribute are defined by the W3C XML Schema int datatype.
custSzY (Fixed Height Override)	Specifies a fixed height override for a shape. The possible values for this attribute are defined by the W3C XML Schema int datatype.
custT (Text	Specifies if the text has been customized which allows layout to ignore automatic

Attributes	Description
Changed)	formatting options available to the text. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
loCatId (Current Diagram Category)	Specifies the current identifier of the layout category applied to the diagram. The possible values for this attribute are defined by the W3C XML Schema string datatype.
loTypeId (Current Diagram Type)	Specifies the identifier for the layout currently applied to the diagram. The possible values for this attribute are defined by the W3C XML Schema string datatype.
phldr (Placeholder)	Indicates that the point is a placeholder or sample item. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
phldrT (Placeholder Text)	The text used for display in the element if the placeholder flag is set to `true`. If this property is not set then the default placeholder text is used. The possible values for this attribute are defined by the W3C XML Schema string datatype.
presAssocID (Presentation Element Identifier)	The point associated with this presentation element. This identifier is used together with presName to create a unique key for presentation element indexing. The possible values for this attribute are defined by the ST_ModelId simple type (§21.4.7.43).
presName (Presentation Name)	The layout node name of this presentation element. This name is used together with presAssocID to create a unique key for presentation element indexing. The possible values for this attribute are defined by the W3C XML Schema string datatype.
presStyleCnt (Presentation Style Count)	Specifies the layout node style count of this presentation element. The possible values for this attribute are defined by the W3C XML Schema int datatype.
presStyleIdx (Presentation Style Index)	Specifies the layout node style index of this presentation element. The possible values for this attribute are defined by the W3C XML Schema int datatype.
presStyleLbl (Presentation Style Label)	Specifies the layout node style label of this presentation element. The possible values for this attribute are defined by the W3C XML Schema string datatype.
qsCatId (Current	Specifies the identifier of the category of the currently applied quick style.

Attributes	Description
Style Category)	The possible values for this attribute are defined by the W3C XML Schema string datatype.
qsTypeId (Current Style Type)	Specifies the identifier of the currently applied quick style. 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_ElemPropSet) is located in §A.5.3. *end note*]

21.4.3.5 pt (Point)

This element defines a point in DiagramML. A point in DiagramML is defined to hold data associated with a particular point or node in a diagram. Transitions between nodes in a diagram along with the nodes themselves are defined as different types of points. A point is not only responsible for holding the data associated with a node in a diagram, but also for holding customization properties made to the text and shape associated with the particular node.

[*Example*: Consider the following example of a pt in DiagramML:

```
<pt modelId="{C6A8900D-3F1B-4F1D-A514-4E8BDD964568}">
    <prSet phldrT="[Text]"/>
    <spPr/>
    <t>
          <bodyPr/>
          <lstStyle/>
          <p>
                <r>
                      <rPr lang="en-US" smtClean="0"/>
                      <t>Text 2</t>
                </r>
                <endParaRPr lang="en-US" dirty="0"/>
          </p>
    </t>
</pt>
```

In this example we define a point which holds the data associated with a node in a diagram. The actual text in the diagram is defined in the text body, t, tag and consists of the string "Text 2". There are no overrides made to the shape properties and placeholder text defined for this node when there is no text body present. *end example*]

Attributes	Description

cxnId (Connection Identifier)	The model identifier of the connection that represents the transition node.
	
	[*Example*: Consider the following example of a cxnId:
	
	`<dgm:pt modelId="5" type="parTrans" cxnId="9">`
	`  <dgm:prSet/>`
	`  <dgm:spPr/>`
	`</dgm:pt>`
	
	In this example we define the connection related to this point to reference connection 9. *end example*]
	
	The possible values for this attribute are defined by the ST_ModelId simple type (§21.4.7.43).

modelId (Model Identifier)	The unique identifier of the element within the data model. This identifier should be unique only relative to the containing data model.
	
	[*Example*: Consider the following example of a cxnId:
	
	`<dgm:pt modelId="5" type="parTrans" cxnId="9">`
	`  <dgm:prSet/>`
	`  <dgm:spPr/>`
	`</dgm:pt>`
	
	In this example we define the point type is to be 5. *end example*]
	
	The possible values for this attribute are defined by the ST_ModelId simple type (§21.4.7.43).

type (Point Type)	The type of point.
	
	[*Example*: Consider the following example of a cxnId:
	
	`<dgm:pt modelId="5" type="parTrans" cxnId="9">`
	`  <dgm:prSet/>`
	`  <dgm:spPr/>`
	`</dgm:pt>`
	
	In this example the point type is defined as a parTrans point type. *end example*]
	
	The possible values for this attribute are defined by the ST_PtType simple type (§21.4.7.51).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Pt) is located in §A.5.3. *end note*]

21.4.3.6 ptLst (Point List)

This element simply holds a list of points within the data model.

[*Example*: Consider the following example of a very simple point list in DiagramML:

```
<dgm:ptLst>
    <dgm:pt modelId="0" type="doc"/>
    <dgm:pt modelId="1"/>
    <dgm:pt modelId="2"/>
    <dgm:pt modelId="3"/>
    <dgm:pt modelId="4"/>
    <dgm:pt modelId="5"/>
    <dgm:pt modelId="6"/>
</dgm:ptLst>
```

In this example we define a single document type point and five node type points. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_PtList) is located in §A.5.3. *end note*]

21.4.3.7 spPr (Shape Properties)

This element specifies the properties for a single shape in a diagram's data, as defined using DrawingML child elements.

Attributes	Description
bwMode (Black and White Mode) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeProperties) is located in §A.4.1. *end note*]

21.4.3.8 t (Text Body)

Text body containing the default body, paragraph and character properties. There should be a signle paragraph and no text runs. Any runs in the first paragraph and paragraphs in addition to the first are ignored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBody) is located in §A.4.1. *end note*]

21.4.3.9 whole (Whole E2O Formatting)

Formatting that applies to the entire diagram object, and not just the background, includes line and effect properties.

[*Note*: The W3C XML Schema definition of this element's content model (CT_WholeE2oFormatting) is located in §A.4.1. *end note*]

21.4.4 Color Information

This section defines the coloring information that is to be associated with a diagram.

21.4.4.1 cat (Color Transform Category)

This element specifies the category in the user interface that a color transform is to be displayed within.

[*Example*: Consider the following example of a cat in DiagramML:

```
<dgm:cat type="mainScheme" pri="10300"/>
```

In this example we see a cat defined with a category type and priority. *end example*]

Attributes	Description
pri (Priority)	The priority within the category for this color variation determines the order in which it displays in the user interface. The lower numbers are to be displayed at the beginning of the list. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
type (Category Type)	The category type used to organize the color transforms in the user interface. The possible values for this attribute are defined by the W3C XML Schema anyURI datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CTCategory) is located in §A.5.3. *end note*]

21.4.4.2 catLst (Color Transform Category List)

This element defines a list of color transform categories. This list can be used to populate user interface components which could separate color transforms into categories.

[*Example*: Consider the following example of a catLst in DiagramML:

```
<dgm:catLst>
```

```
      <dgm:cat type="mainScheme" pri="10300"/>
  </dgm:catLst>
```

In this example we see a catLst defined which holds a single *color transform category* (§21.4.4.1). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_CTCategories) is located in §A.5.3. *end note*]

21.4.4.3 colorsDef (Color Transform Definitions)

This element is the root element for color transforms. Held within this element are all of the available color transforms themselves along with other elements and attributes associated with defining the general color transform properties and attributes.

[*Example*: Consider the following example of a colorsDef in DiagramML:

```
<dgm:colorsDef
xmlns:dgm="http://schemas.openxmlformats.org/drawingml/2006/diagram"
xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
uniqueId="urn:microsoft.com/office/officeart/2005/8/colors/accent0_3"
minVer="12.0">
    <dgm:title lang="" val=""/>
    <dgm:desc lang="" val=""/>
    <dgm:catLst>
          <dgm:cat type="mainScheme" pri="10300"/>
    </dgm:catLst>
    <dgm:styleLbl name="node0">
          …
    </dgm:styleLbl>
    …
</dgm:colorsDef>
```

In this example we see a sampling of a colorsDef being defined with a number of styleLbl elements held within the colorsDef. *end example*]

Attributes	Description

minVer (Minimum Version)	The minimum product version that can support this color transform.
	
	[*Example*: Consider the following example:
	
	`<colorsDef uniqueId="urn:colors/accent0_3" minVer="12.0">`
	`…`
	`</colorsDef>`

Attributes	Description
	In this example we see the minVer set to `12.0`. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

uniqueId (Unique ID)	A unique id associated with the color transform definition.
	
	[*Example*: Consider the following example:
	
	`<colorsDef uniqueId="urn:colors/accent0_3" minVer="12.0">`
	`…`
	`</colorsDef>`
	
	In this example we see the uniqueId set to `urn:colors/accent0_3`. *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_ColorTransform) is located in §A.5.3. *end note*]

21.4.4.4 colorsDefHdr (Color Transform Definition Header)

This element specifies header information associated with a color transform definition. The header information is used by an application to preprocess required data in order to help with possible performance concerns associated with an initial full load of a color transform definition.

[*Example*: Consider the following example of a colorsDefHdr within DiagramML:

```
<colorsDefHdr uniqueId="urn:colors/accent0_1">
    <title val="Main 1" />
    <desc val="" />
    <catLst>
        <cat type="mainScheme" pri="10100" />
    </catLst>
</colorsDefHdr>
```

In this example we see a color transform definition header which defines a title and category for a set of color transforms. *end example*]

Attributes	Description
minVer (Minimum Version)	The minimum product version that can support the associated color transform definition. The possible values for this attribute are defined by the W3C XML Schema string datatype.

Attributes	Description
resId (Resource ID)	This attribute is the id which associates this header to the actual color transform definition. The possible values for this attribute are defined by the W3C XML Schema int datatype.
uniqueId (Unique ID)	This attribute defines a unique identifier for the associated color transform definition. 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_ColorTransformHeader) is located in §A.5.3. *end note*]

21.4.4.5 colorsDefHdrLst (Color Transform Header List)

This element is simply a list of color transform definition headers and is used to consolidate multiple headers in a group.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorTransformHeaderLst) is located in §A.5.3. *end note*]

21.4.4.6 desc (Description)

This element holds a description for a color definition. The description can be used to describe the qualities associated with a particular color transform definition.

Attributes	Description
lang (Language)	The natural language of the color transform definition. The possible values for this attribute are defined by the W3C XML Schema string datatype.
val (Description Value)	The string which is used as the description of the color transform definition. 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_CTDescription) is located in §A.5.3. *end note*]

21.4.4.7 effectClrLst (Effect Color List)

This element defines a list of colors applied to effects within a color transform.

[*Example*: Consider the following example of an effectClrLst in DiagramML:


```
<dgm:effectClrLst meth="repeat">
    <a:schemeClr val="dk2">
          <a:tint val="60000"/>
    </a:schemeClr>
</dgm:effectClrLst>
```

In this example we see a single color defined in the effectClrLst, more specifically we see a scheme color being utilized with a tint applied to the color. *end example*]

Attributes	Description

hueDir (Hue Direction)	The direction around the color wheel the hue shift (if defined) occurs.
	
	[*Example*: Consider the following example of a hueDir in use:
	
	`<dgm:lnClrLst hueDir="cw" meth="repeat">`
	`  <a:schemeClr val="dk2">`
	`  <a:tint val="60000"/>`
	`  </a:schemeClr>`
	`</dgm:lnClrLst>`
	
	In this example an lnClrLst is defined with a hue direction defined as clockwise. *end example*]
	
	The possible values for this attribute are defined by the ST_HueDir simple type (§21.4.7.38).

meth (Color Application Method Type)	The method used to apply the color transform.
	
	[*Example*: Consider the following example of a meth in use:
	
	`<dgm:lnClrLst hueDir="cw" meth="repeat">`
	`  <a:schemeClr val="dk2">`
	`  <a:tint val="60000"/>`
	`  </a:schemeClr>`
	`</dgm:lnClrLst>`
	
	In this example and lnClrLst is defined using the `repeat` color application method. *end example*]
	
	The possible values for this attribute are defined by the ST_ClrAppMethod simple type (§21.4.7.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Colors) is located in §A.5.3. *end note*]

21.4.4.8 fillClrLst (Fill Color List)

This element defines a list of colors which are used as fill colors in the color transform. The fill colors define the color of the nodes in a diagram.

[*Example*: Consider the following example of a fillClrLst in DiagramML:

```
<dgm:fillClrLst meth="repeat">
     <a:schemeClr val="dk2"/>
</dgm:fillClrLst>
```

In this example the fillClrList contains a single scheme color and is utilizing the repeat method for color application. *end example*]

Attributes	Description

hueDir (Hue Direction)	The direction around the color wheel the hue shift (if defined) occurs.
	
	[*Example*: Consider the following example of a hueDir in use:
	
	`<dgm:lnClrLst hueDir="cw" meth="repeat">`
	`  <a:schemeClr val="dk2">`
	`  <a:tint val="60000"/>`
	`  </a:schemeClr>`
	`</dgm:lnClrLst>`
	
	In this example an lnClrLst is defined with a hue direction defined as clockwise. *end example*]
	
	The possible values for this attribute are defined by the ST_HueDir simple type (§21.4.7.38).

meth (Color Application Method Type)	The method used to apply the color transform.
	
	[*Example*: Consider the following example of a meth in use:
	
	`<dgm:lnClrLst hueDir="cw" meth="repeat">`
	`  <a:schemeClr val="dk2">`
	`  <a:tint val="60000"/>`
	`  </a:schemeClr>`
	`</dgm:lnClrLst>`
	
	In this example and lnClrLst is defined using the `repeat` color application method. *end example*]
	
	The possible values for this attribute are defined by the ST_ClrAppMethod simple type (§21.4.7.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Colors) is located in §A.5.3. *end note*]

21.4.4.9 linClrLst (Line Color List)

This element defines a list of colors which are used as line colors in the color transform. The line colors define the color of the lines used on a given node in a diagram

[*Example*: Consider the following example of a lnClrLst in DiagramML:

```
<dgm:lnClrLst meth="repeat">
     <a:schemeClr val="dk2"/>
</dgm:lnClrLst>
```

In this example the lnClrList contains a single scheme color and is utilizing the repeat method for color application. *end example*]

Attributes	Description

hueDir (Hue Direction)	The direction around the color wheel the hue shift (if defined) occurs.
	
	[*Example*: Consider the following example of a hueDir in use:
	
	`<dgm:lnClrLst hueDir="cw" meth="repeat">`
	`  <a:schemeClr val="dk2">`
	`  <a:tint val="60000"/>`
	`  </a:schemeClr>`
	`</dgm:lnClrLst>`
	
	In this example an lnClrLst is defined with a hue direction defined as clockwise. *end example*]
	
	The possible values for this attribute are defined by the ST_HueDir simple type (§21.4.7.38).

meth (Color Application Method Type)	The method used to apply the color transform.
	
	[*Example*: Consider the following example of a meth in use:
	
	`<dgm:lnClrLst hueDir="cw" meth="repeat">`
	`  <a:schemeClr val="dk2">`
	`  <a:tint val="60000"/>`
	`  </a:schemeClr>`
	`</dgm:lnClrLst>`
	
	In this example and lnClrLst is defined using the `repeat` color application method. *end example*]
	
	The possible values for this attribute are defined by the ST_ClrAppMethod simple type

Attributes	Description
	(§21.4.7.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Colors) is located in §A.5.3. *end note*]

21.4.4.10 styleLbl (Style Label)

This element defines a style label. The style label is used to define a color transform that is applied to a given node in a diagram.

[*Example*: Consider the following example of a styleLbl in DiagramML:

```
<dgm:styleLbl name="exampleStyleLabel">
    <dgm:fillClrLst>
          <a:schemeClr val="accent2"/>
          <a:schemeClr val="accent3"/>
    </dgm:fillClrLst>
    <dgm:linClrLst meth="repeat">
          <a:schemeClr val="lt1"/>
    </dgm:linClrLst>
    <dgm:effectClrLst/>
    <dgm:txLinClrLst/>
    <dgm:txFillClrLst/>
    <dgm:txEffectClrLst/>
</dgm:styleLbl>
```

In this example we see a style label defined in its entirety. This style label can be used on a layout node in order to define the color transform that is to be applied to the node. *end example*]

Attributes	Description

name (Name)	A name given to the style label. This name can be referenced by layout nodes in order to apply the style label to the layout node.
	
	[*Example*: Consider the following example of a styleLbl in DiagramML:
	`<dgm:styleLbl name="exampleStyleLabel">`
	…
	`</dgm:styleLbl>`
	
	In this example we see a style label defined with the name `exampleStyleLabel` defined. *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_CTStyleLabel) is located in §A.5.3. *end note*]

21.4.4.11 title (Title)

The name or title given to the color definition header.

Attributes	Description
lang (Language)	The natural language of the title or description of a color transform definition. The possible values for this attribute are defined by the W3C XML Schema string datatype.
val (Description Value)	A string used for a description of a color transform definition. 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_CTName) is located in §A.5.3. *end note*]

21.4.4.12 txEffectClrLst (Text Effect Color List)

This element defines a list of colors which are used as text effect colors in the color transform. The text effect colors define the color of the text effects used on a given node in a diagram

[*Example*: Consider the following example of a txEffectClrLst in DiagramML:

```
<dgm:txEffectClrLst meth="repeat">
     <a:schemeClr val="dk2"/>
</dgm:txEffectClrLst>
```

In this example the txEffectClrLst contains a single scheme color and is utilizing the repeat method for color application. *end example*]

Attributes	Description

hueDir (Hue Direction)	The direction around the color wheel the hue shift (if defined) occurs.
	
	[*Example*: Consider the following example of a hueDir in use:
	
	`<dgm:lnClrLst hueDir="cw" meth="repeat">`
	`  <a:schemeClr val="dk2">`
	`  <a:tint val="60000"/>`
	`  </a:schemeClr>`
	`</dgm:lnClrLst>`
	
	In this example an lnClrLst is defined with a hue direction defined as clockwise. *end*

Attributes	Description
	example] The possible values for this attribute are defined by the ST_HueDir simple type (§21.4.7.38).

meth (Color Application Method Type)	The method used to apply the color transform.
	
	[*Example*: Consider the following example of a meth in use:
	
	`<dgm:lnClrLst hueDir="cw" meth="repeat">`
	`<a:schemeClr val="dk2">`
	`<a:tint val="60000"/>`
	`</a:schemeClr>`
	`</dgm:lnClrLst>`
	
	In this example and lnClrLst is defined using the `repeat` color application method. *end example*]
	
	The possible values for this attribute are defined by the ST_ClrAppMethod simple type (§21.4.7.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Colors) is located in §A.5.3. *end note*]

21.4.4.13 txFillClrLst (Text Fill Color List)

This element defines a list of colors which are used as text colors in the color transform. The text colors define the color of the text used in a given node in a diagram

[*Example*: Consider the following example of a txFillClrLst in DiagramML:

```
<dgm:txFillClrLst meth="repeat">
     <a:schemeClr val="dk2"/>
</dgm:txFillClrLst>
```

In this example the txFillClrLst contains a single scheme color and is utilizing the repeat method for color application. *end example*]

Attributes	Description

hueDir (Hue Direction)	The direction around the color wheel the hue shift (if defined) occurs.
	
	[*Example*: Consider the following example of a hueDir in use:
	
	`<dgm:lnClrLst hueDir="cw" meth="repeat">`
	`<a:schemeClr val="dk2">`
	`<a:tint val="60000"/>`

Attributes	Description
	</a:schemeClr> </dgm:lnClrLst> In this example an lnClrLst is defined with a hue direction defined as clockwise. *end example*] The possible values for this attribute are defined by the ST_HueDir simple type (§21.4.7.38).
meth (Color Application Method Type)	The method used to apply the color transform. [*Example*: Consider the following example of a meth in use: <dgm:lnClrLst hueDir="cw" meth="repeat"> <a:schemeClr val="dk2"> <a:tint val="60000"/> </a:schemeClr> </dgm:lnClrLst> In this example and lnClrLst is defined using the repeat color application method. *end example*] The possible values for this attribute are defined by the ST_ClrAppMethod simple type (§21.4.7.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Colors) is located in §A.5.3. *end note*]

21.4.4.14 txLinClrLst (Text Line Color List)

This element defines a list of colors which are used as text line colors in the color transform. The text line colors define the color of the line on text used in a given node in a diagram

[*Example*: Consider the following example of a txLinClrLst in DiagramML:

```
<dgm:txLinClrLst meth="repeat">
     <a:schemeClr val="dk2"/>
</dgm:txLinClrLst>
```

In this example the txLinClrLst contains a single scheme color and is utilizing the repeat method for color application. *end example*]

Attributes	Description
hueDir (Hue Direction)	The direction around the color wheel the hue shift (if defined) occurs. [*Example*: Consider the following example of a hueDir in use:

Attributes	Description
	<dgm:lnClrLst hueDir="cw" meth="repeat"> <a:schemeClr val="dk2"> <a:tint val="60000"/> </a:schemeClr> </dgm:lnClrLst> In this example an lnClrLst is defined with a hue direction defined as clockwise. *end example*] The possible values for this attribute are defined by the ST_HueDir simple type (§21.4.7.38).
meth (Color Application Method Type)	The method used to apply the color transform. [*Example*: Consider the following example of a meth in use: <dgm:lnClrLst hueDir="cw" meth="repeat"> <a:schemeClr val="dk2"> <a:tint val="60000"/> </a:schemeClr> </dgm:lnClrLst> In this example and lnClrLst is defined using the repeat color application method. *end example*] The possible values for this attribute are defined by the ST_ClrAppMethod simple type (§21.4.7.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Colors) is located in §A.5.3. *end note*]

21.4.5 Style Definitions

This section describes the styling information to be associated with a diagram.

21.4.5.1 cat (Category)

The category in the user interface where this quick style displays in the user interface.

[*Example*: Consider the following example of a cat in use in DiagramML:

```
<catLst>
    <cat type="3D" pri="11100"/>
</catLst>
```

In this example we see a 3D category type being defined with a priority of 11100. *end example*]

Attributes	Description
pri (Priority)	The priority within the category for this style determines the order in which it displays in the user interface. Lower numbers are displayed at the beginning of the list. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
type (Category Type)	Category type. This is used to organize the quick style in the user interface. The possible values for this attribute are defined by the W3C XML Schema anyURI datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SDCategory) is located in §A.5.3. *end note*]

21.4.5.2 catLst (Category List)

This element is simply a list of categories.

[*Example*: Consider the following example of a catLst in use in DiagramML:

```
<catLst>
    <cat type="Simple" pri="10000"/>
    <cat type="3D" pri="11100"/>
</catLst>
```

In this example two categories defined in the category list. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SDCategories) is located in §A.5.3. *end note*]

21.4.5.3 desc (Style Label Description)

This element defines a description for a style label definition. The description is simply a string describing the characteristics of the style label definition.

[*Example*: Consider the following example of the desc element in DiagramML:

```
<desc lang="" val="3-D Style 1"/>
```

In this example we define the description to be `3-D Style 1`. *end example*]

Attributes	Description
lang (Natural Language)	The natural language of the title or description of this quick style. The possible values for this attribute are defined by the W3C XML Schema string datatype.

Attributes	Description
val (Description Value)	The string used for the 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_SDDescription) is located in §A.5.3. *end note*]

21.4.5.4 presLayoutVars (Presentation Layout Variables)

This element specified the layout property set. This set of properties determine different aspects concerning the layout of a diagram. All of the elements associated with enabling or disabling aspects of the user interface are also defined here.

[*Note*: The W3C XML Schema definition of this element's content model (CT_LayoutVariablePropertySet) is located in §A.5.3. *end note*]

21.4.5.5 scene3d (3-D Scene)

The 3-D scene which consists of a camera, a light rig, and an optional backdrop to catch shadows.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Scene3D) is located in §A.4.1. *end note*]

21.4.5.6 sp3d (3-D Shape Properties)

A set of 3-D properties which a shape can contain.

Attributes	Description
contourW (Contour Width) Namespace: http://purl.oclc.org/ooxml/drawingml/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>`

Attributes	Description
	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 (§20.1.10.41).
extrusionH (Extrusion Height) Namespace: http://purl.oclc.org/ooxml/drawingml/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 (§20.1.10.41).
prstMaterial (Preset Material Type) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Defines the preset material which 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

Attributes	Description
	type (§20.1.10.49).
z (Shape Depth) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Defines the z coordinate for the 3D shape. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Shape3D) is located in §A.4.1. *end note*]

21.4.5.7 styleDef (Style Definition)

This element is the root tag for a style definition.

[*Example*: Consider the following example of a styleDef in DiagramML:

```
<dgm:styleDef
xmlns:dgm="http://schemas.openxmlformats.org/drawingml/2006/diagram"
xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
uniqueId="urn:microsoft.com/office/officeart/2005/8/quickstyle/3d1"
minVer="12.0">
    <dgm:title lang="" val="3-D Style 1"/>
    <dgm:desc lang="" val="3-D Style 1"/>
    <dgm:catLst>
          <dgm:cat type="3D" pri="11100"/>
    </dgm:catLst>
    <dgm:scene3d>
          <a:camera prst="orthographicFront"/>
          <a:lightRig rig="threePt" dir="t"/>
    </dgm:scene3d>
    <dgm:style>
                    …
    </dgm:style>
    <dgm:styleLbl name="node0">
          <dgm:scene3d>
                <a:camera prst="orthographicFront"/>
                <a:lightRig rig="flat" dir="t"/>
          </dgm:scene3d>
          <dgm:sp3d prstMaterial="flat">
                <a:bevelT w="120900" h="88900"/>
                <a:bevelB w="88900" h="31750" prst="angle"/>
          </dgm:sp3d>
```



```
            <dgm:txPr/>
            <dgm:style>
                  <a:lnRef idx="0">
                        <a:scrgbClr r="0" g="0" b="0"/>
                  </a:lnRef>
                  <a:fillRef idx="3">
                        <a:scrgbClr r="0" g="0" b="0"/>
                  </a:fillRef>
                  <a:effectRef idx="2">
                        <a:scrgbClr r="0" g="0" b="0"/>
                  </a:effectRef>
                  <a:fontRef idx="minor">
                        <a:schemeClr val="lt1"/>
                  </a:fontRef>
            </dgm:style>
      </dgm:styleLbl>
…
</dgm:styleDef>
```

In this example we see a styleDef being defined along with many properties. *end example*]

Attributes	Description

minVer (Minimum Version)	The minimum product version that can support this quick style.
	
	[*Example*: Consider the following example of a styleDef in DiagramML:
	
	`<styleDef uniqueId="urn:quickstyle/3d1" minVer="12.0">`
	
	…
	
	`</styleDef>`
	
	In this example we see the `minVer` defined to be `12.0`. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

uniqueId (Unique Style ID)	Unique ID that identifies a style. [*Example*: Consider the following example of a styleDef in DiagramML: `<styleDef uniqueId="urn:quickstyle/3d1" minVer="12.0">` …

Attributes	Description
	</styleDef> In this example we see the uniqueId defined to be urn:quickstyle/3d1. *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_StyleDefinition) is located in §A.5.3. *end note*]

21.4.5.8 styleDefHdr (Style Definition Header)

This element specifies header information associated with a style definition. The header information is used by an application to preprocess required data in order to help with possible performance concerns associated with an initial full load of a color transform definition.

[*Example*: Consider the following example of a styleDefHdr element within DiagramML:

```
<styleDefHdr uniqueId="urn:quicktyle/3d1">
    <title val="3D" />
    <desc val="" />
    <catLst>
        <cat type="3D" pri="10100" />
    </catLst>
</ styleDefHdr >
```

In this example we see a style definition header which defines a title and category for a set of style definitions. *end example*]

Attributes	Description
minVer (Minimum Version)	The minimum product version that can support this quick style. The possible values for this attribute are defined by the W3C XML Schema string datatype.
resId (Resource ID)	This attribute is the id which associates this header to the actual style definition part. The possible values for this attribute are defined by the W3C XML Schema int datatype.
uniqueId (Unique Style ID)	This attribute defines a unique identifier for the associated style definition. 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_StyleDefinitionHeader) is located in §A.5.3. *end note*]

21.4.5.9 styleDefHdrLst (List of Style Definition Headers)

This element is simply a list of style definition headers and is used to consolidate multiple headers into one group.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StyleDefinitionHeaderLst) is located in §A.5.3. *end note*]

21.4.5.10 styleLbl (Style Label)

This element defines the actual style which is applied to a node in a diagram The style is referenced from within layout node. The style label contains formatting (without defining color) such as the 3D properties and text properties associated with a shape.

[*Example*: Consider the following example of a styleLbl in DiagramML:

```
<styleLbl name="node0">
    <scene3d>
          <camera prst="orthographicFront"/>
          <lightRig rig="flat" dir="t"/>
    </scene3d>
    <sp3d prstMaterial="flat">
          <bevelT w="120900" h="88900"/>
          <bevelB w="88900" h="31750" prst="angle"/>
    </sp3d>
    <txPr/>
    <style>
          <lnRef idx="0">
                <scrgbClr r="0" g="0" b="0"/>
          </lnRef>
          <fillRef idx="3">
                <scrgbClr r="0" g="0" b="0"/>
          </fillRef>
          <effectRef idx="2">
                <scrgbClr r="0" g="0" b="0"/>
          </effectRef>
          <fontRef idx="minor">
                <schemeClr val="lt1"/>
          </fontRef>
    </style>
</styleLbl>
```

In this example we see a styleLbl defined which sets 3D properties for the scene, shape 3D properties, line, fill, effect and font properties. *end example*]

Attributes	Description
name (Style Name)	The name of the style. This apears as the tooltip in the user interface. 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_StyleLabel) is located in §A.5.3. *end note*]

21.4.5.11 title (Title)

This element defines the title given to a style definition header. The title is simply a name for the style definition.

[*Example*: Consider the following example of title being used in DiagramML:

```
<styleDefHdr uniqueId="urn:quickstyle/3d1" minVer="12.0">
     <title val="My Title"/>
     <desc val="My Description"/>

     …

</styleDefHdr>
```

In this example we see the title being set to `My Title`. *end example*]

Attributes	Description
lang (Natural Language)	The natural language of the title or description of this quick style. The possible values for this attribute are defined by the W3C XML Schema string datatype.
val (Description Value)	The string used for the 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_SDName) is located in §A.5.3. *end note*]

21.4.5.12 txPr (Text Properties)

This element defines special text formatting that can be applied to text through a style label.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextProps) is located in §A.5.3. *end note*]

21.4.6 Layout Definition

This section specifies the node layout information to be associated with a diagram.

21.4.6.1 animLvl (Level Animation)

This variable is used to indicate the animate by level string which is displayed to a user in the user interface.

[*Example*: Consider the following example of animLvl in use in DiagramML:

```
<varLst>
     <chMax val="1" />
     <dir val="norm" />
     <animLvl val="ctr" />
     <resizeHandles val="exact" />
</varLst>
```

In this example we see that the animLvl is set to `ctr`. This is being defined in a radial type diagram which allows the user to specify that animation is to start at the center of the diagram. *end example*]

Attributes	Description
val (Level Animation Value)	This attribute indicates the string to use for level animation in the user interface. The possible values for this attribute are defined by the ST_AnimLvlStr simple type (§21.4.7.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AnimLvl) is located in §A.5.3. *end note*]

21.4.6.2 animOne (One by One Animation String)

This variable is used to indicate the string to use for one-by-one animation in the user interface. This is used primarily when defining hierarchical diagrams to specify different ways animations applies to different levels of the diagram.

[*Example*: Consider the following example of animOne used in a hierarchical type diagram:

```
<varLst>
     <chPref val="1" />
     <dir val="norm" />
     <animOne val="branch" />
     <animLvl val="lvl" />
     <resizeHandles val="exact" />
</varLst>
```

In this example we see that the animOne element is defined to animate the diagram per branch. *end example*]

Attributes	Description
val (One By One Animation Value)	Specifies the type of one-by-one animation to use for a diagram. The possible values for this attribute are defined by the ST_AnimOneStr simple type (§21.4.7.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AnimOne) is located in §A.5.3. *end note*]

21.4.6.3 bulletEnabled (Show Insert Bullet)

This element is used to indicate when to enable the 'Insert Bullet' button in the user interface.

[*Example*: Consider the following example of bulletEnabled in DiagramML:

```
<varLst>
    <bulletEnabled val="true" />
</varLst>
```

In this example we see that the insert button in the user interface is to be enabled when the focus is within the containing layout node. *end example*]

Attributes	Description
val (Show Insert Bullet Value)	This attribute is used to indicate when to enable the 'Insert Bullet' button. A value of `true` enables the insert bullet button. 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_BulletEnabled) is located in §A.5.3. *end note*]

21.4.6.4 chMax (Maximum Children)

This element is used to indicate when to enable and disable the user interface components associated with adding a new shape to a diagram. This element defines a max number of nodes a diagram can support through the user interface directly.

[*Example*: Consider the following example of chMax usage in DiagramML:

```
<varLst>
    <chMax val="5"/>
    <dir val="norm"/>
```

```
    <resizeHandles val="exact" />
</varLst>
```

In this example we define the user interface to only be enabled to insert five nodes. *end example*]

Attributes	Description
val (Maximum Children Value)	This attribute indicates the maximum number of children the node can have before the user interface should be disabled. A value of -1 indicates an infinite number of children. Default value is -1. The possible values for this attribute are defined by the ST_NodeCount simple type (§21.4.7.44).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChildMax) is located in §A.5.3. *end note*]

21.4.6.5 chPref (Preferred Number of Children)

This variable indicates the number of children that the current node prefers to have. This determines what the next action of the 'Add Shape' button should be in the user interface.

[*Example*: Consider the following example of chPref being used in DiagramML:

```
<varLst>
    <chMax val="3" />
    <chPref val="1" />
    <dir val="norm" />
    <animLvl val="lvl" />
    <resizeHandles val="rel" />
</varLst>
```

In this example, chPref is set to a single node and the user interface disables after a single node has been inserted. *end example*]

Attributes	Description
val (Preferred Number of CHildren Value)	This attribute indicates the number of children that the current node prefers to have. This determines what the next action of the 'Add Shape' button should be. A value of -1 indicates an infinite number of children. Default value is -1. The possible values for this attribute are defined by the ST_NodeCount simple type (§21.4.7.44).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChildPref) is located in §A.5.3. *end note*]

21.4.6.6 dir (Diagram Direction)

This element indicates whether the diagram should switch direction. This element provides the ability to define different behavior for diagrams considering LTR or RTL directions.

Attributes	Description
val (Diagram Direction Value)	This variable indicates whether the diagram should switch direction. The possible values for this attribute are defined by the ST_Direction simple type (§21.4.7.26).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Direction) is located in §A.5.3. *end note*]

21.4.6.7 hierBranch (Organization Chart Branch Style)

This element defines the layout style of a branch in an organizational chart.

[*Example*: Consider the following example of hierBranch being used in DiagramML:

```
<varLst>
    <hierBranch val="init" />
</varLst>
```

In this example the value of hierBranch is defined as `init` which is a kind of not set state, or initial state. *end example*]

Attributes	Description
val (Organization Chart Branch Style Value)	The value of this attribute indicates the layout style of a branch in an organization chart. The default value is `std`. The possible values for this attribute are defined by the ST_HierBranchStyle simple type (§21.4.7.37).

[*Note*: The W3C XML Schema definition of this element's content model (CT_HierBranchStyle) is located in §A.5.3. *end note*]

21.4.6.8 orgChart (Show Organization Chart User Interface)

This element is used to indicate when to show user interface controls specifically associated with organizational charts such as being able to add an assistant to a selected node.

[*Example*: Consider the following example of orgChart used in DiagramML:

```
<varLst>
    <orgChart val="true" />
```

```
      <chPref val="1" />
      <dir val="norm" />
      <animOne val="branch" />
      <animLvl val="lvl" />
      <resizeHandles val="rel" />
   </varLst>
```

In this example we set the orgChart value to `true` indicating that the organizational chart specific user interface controls are to be enabled when the containing diagram is used. *end example*]

Attributes	Description
val (Show Organization Chart User Interface Value)	This attribute value specifies when to show the 'Insert Assistant' user interface control and the 'Change Layout' user interface for this diagram. 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_OrgChart) is located in §A.5.3. *end note*]

21.4.7 Simple Types

This is the complete list of simple types dedicated to DrawingML – Diagrams.

21.4.7.1 ST_AlgorithmType (Algorithm Types)

Types of available algorithms.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
composite (Composite)	The composite algorithm specifies the size and position for all child layout nodes. You can use it to create graphics with a predetermined layout or in combination with other algorithms to create more complex shapes.
conn (Connector Algorithm)	The connector algorithm lays out and routes connecting lines, arrows, and shapes between layout nodes.
cycle (Cycle Algorithm)	The cycle algorithm lays out child layout nodes around a circle or portion of a circle using equal angle spacing.
hierChild (Hierarchy Child Algorithm)	The hierarchy child algorithm works with the hierRoot algorithm to create hierarchical tree layouts. This algorithm aligns and positions its child layout nodes in

Enumeration Value	Description
	a linear path under the hierRoot layout node.
hierRoot (Hierarchy Root Algorithm)	The hierarchy root algorithm works with the hierChild algorithm to create hierarchical tree layouts. The hierRoot algorithm aligns and positions the hierRoot layout node in relation to the hierChild layout nodes.
lin (Linear Algorithm)	The linear algorithm lays out child layout nodes along a linear path.
pyra (Pyramid Algorithm)	The pyramid algorithm lays out child layout nodes along a vertical path and works with the trapezoid shape to create a pyramid.
snake (Snake Algorithm)	The snake algorithm lays out child layout nodes along a linear path in two dimensions, allowing the linear flow to continue across multiple rows or columns.
sp (Space Algorithm)	The space algorithm is used to specify a minimum space between other layout nodes or as an indication to do nothing with the layout node's size and position.
tx (Text Algorithm)	The text algorithm sizes text to fit inside a shape and controls its margins and alignment.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AlgorithmType) is located in §A.5.3. *end note*]

21.4.7.2 ST_AnimLvlStr (Animation Level String Definition)

This simple type specifies the possible values for the string that should be displayed by a consumer for level animation of this diagram.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ctr (From Center Animation)	This value specifies that the consumer shall allow "From Center At Once" or "From Center One by One" animation styles for this diagram.
lvl (By Level Animation)	This value specifies that the consumer shall display "By Level" animation types for this diagram.
none (Disable Level At Once)	This value specifies that the consumer shall disable level at once animation.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AnimLvlStr) is located in §A.5.3. *end note*]

21.4.7.3 ST_AnimOneStr (One by One Animation Value Definition)

This simple type defines the possible values for the string to use for one by one animation in the UI. Default value is one.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
branch (By Branch One By One)	This value specifies that the one by one animation string in the user interface should read "By Branch One By One".
none (Disable One-by-One)	This value specifies that the consumer should disable one by one animation.
one (One By One)	This value specifies that the one by one animation string in the user interface should read "One By One".

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AnimOneStr) is located in §A.5.3. *end note*]

21.4.7.4 ST_ArrowheadStyle (Arrowhead Styles)

This simple type defines different arrowhead style types for connectors.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
arr (Arrowhead Present)	Specifies that an arrowhead is to be used on the connector.
auto (Auto)	Specifies that the algorithm defines if an arrowhead is to be used on a connector.
noArr (No Arrowhead)	Specifies no arrowhead is to be used on the connector.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ArrowheadStyle) is located in §A.5.3. *end note*]

21.4.7.5 ST_AutoTextRotation (Auto Text Rotation)

This simple type defines how text rotates within a shape when the shape is rotated by an algorithm during layout.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
grav (Gravity)	Specifies that when the angle of the text hits the threshold of 90 degrees and 180 degrees, the text rotates by 180 degrees.
none (None)	Specifies that text always rotates with the shape.
upr (Upright)	Specifies that when the text angle hits 45, 135, 225, or 315 degree thresholds, then it rotates by negative 90 degrees.

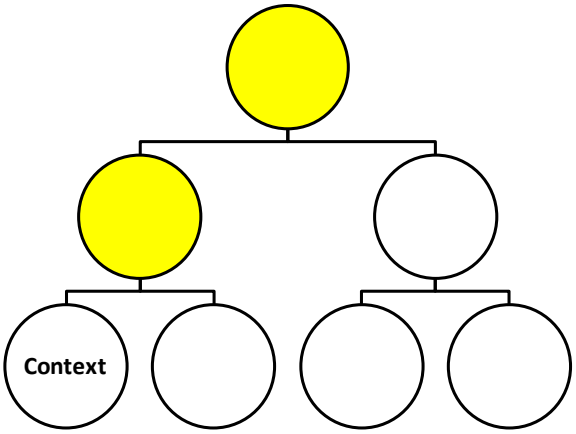
[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AutoTextRotation) is located in §A.5.3. *end note*]

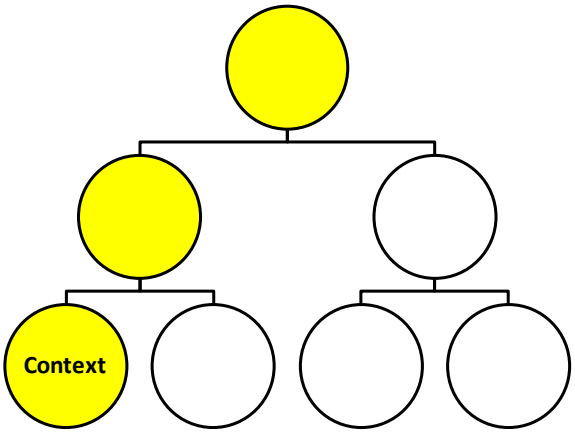
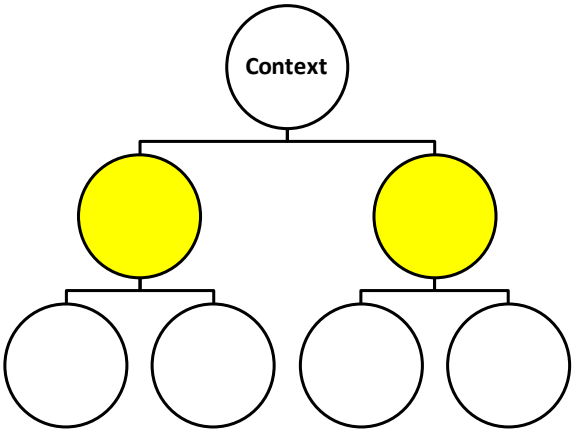
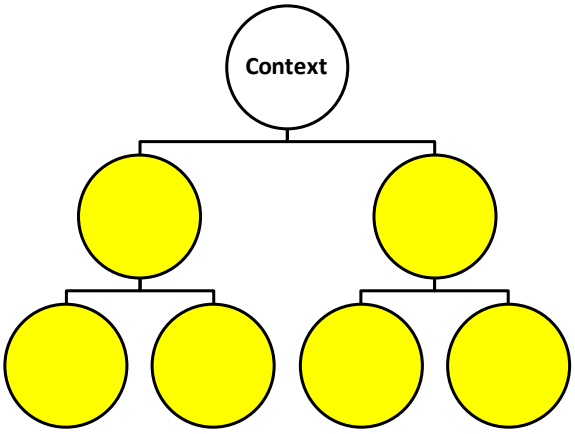
21.4.7.6 ST_AxisType (Axis Type)

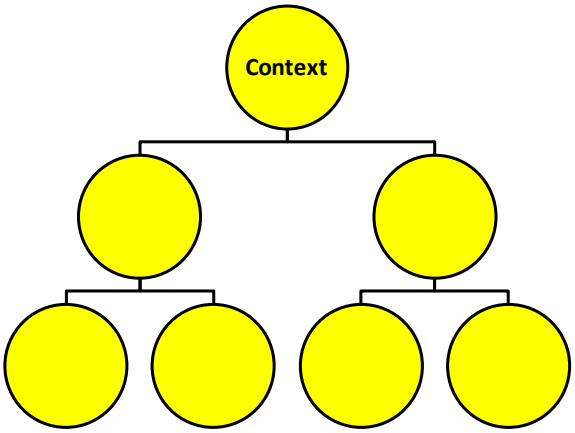
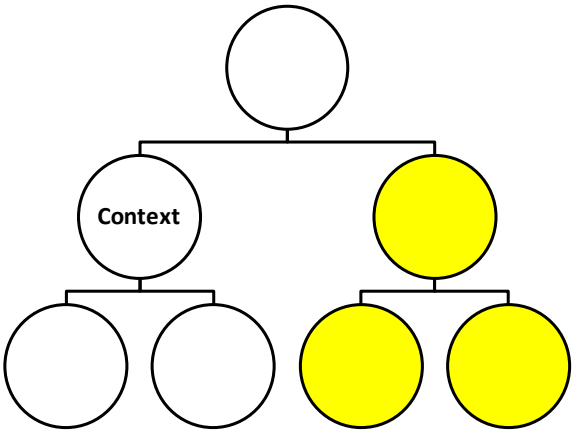
This simple type defines different node sets in relation to the current context node.

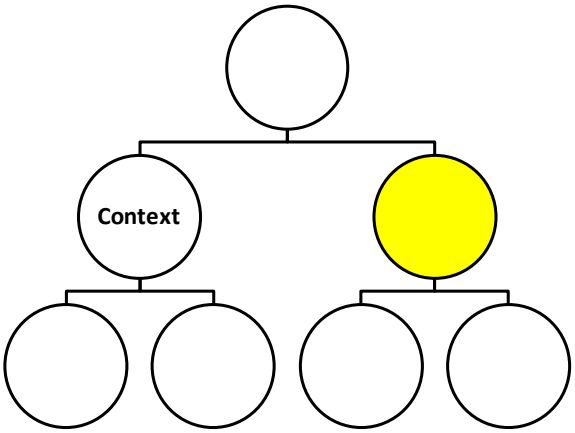
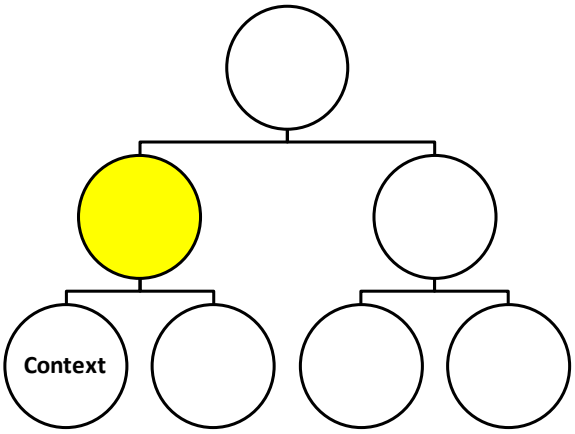
This simple type's contents are a restriction of the W3C XML Schema token datatype.

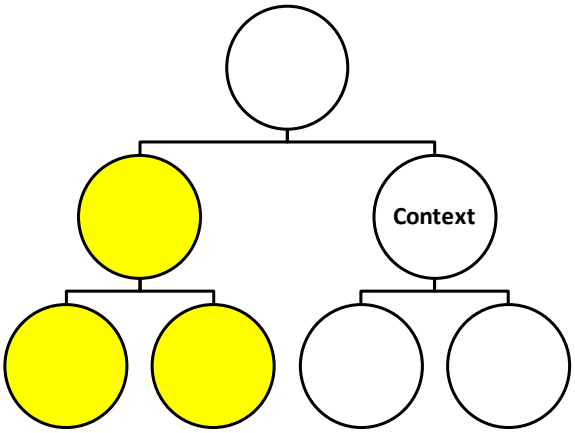
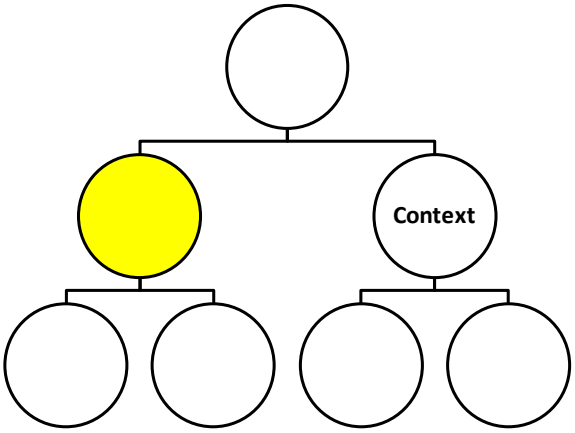
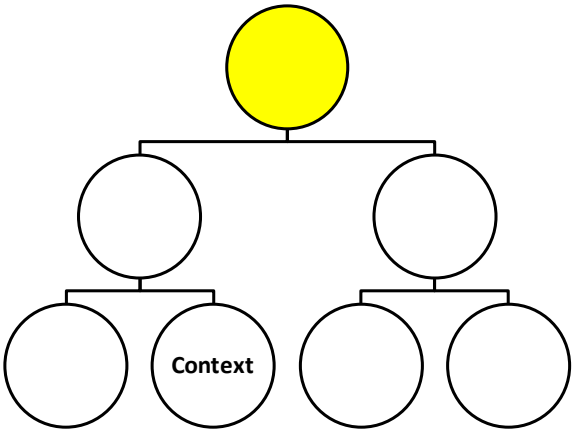
This simple type is restricted to the values listed in the following table:

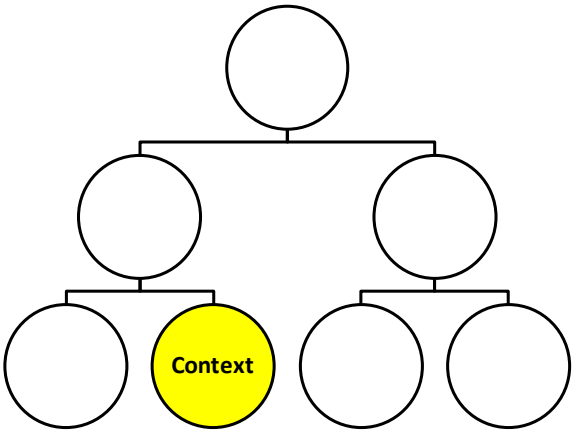
Enumeration Value	Description
ancst (Ancestor)	Specifies a set of nodes between the current context node and the root node, including the root node. [*Example*: Consider the following example  *end example*]
ancstOrSelf (Ancestor or Self)	Specifies a set of nodes between the current context node and the root node, including the root node and the context node. [*Example*: Consider the following example

Enumeration Value	Description
	 *end example*]
ch (Child)	Specifies a set of children of the current context node. [*Example*: Consider the following example  *end example*]
des (Descendant)	Specifies a set of all nodes beneath the current context node. [*Example*: Consider the following example 

Enumeration Value	Description
	end example]
desOrSelf (Descendant or Self)	Specifies a set of all nodes beneath the current context node, including the context node. [*Example*: Consider the following example  *end example*]
follow (Follow)	Specifies the set of nodes which are peers after the context node and all descendants of the peers. [*Example*: Consider the following example  *end example*]
followSib (Follow Sibling)	Specifies the set of nodes which are peers after the context node. [*Example*: Consider the following example

Enumeration Value	Description
	 *end example*]
none (None)	Specifies no node.
par (Parent)	Specifies the parent node. [*Example*: Consider the following example  *end example*]
preced (Preceding)	Specifies the set of nodes which are peers before the context node and all the descendants of the peers. [*Example*: Consider the following example

Enumeration Value	Description
	 *end example*]
precedSib (Preceding Sibling)	Specifies the set of nodes which are peers before the context node. [*Example*: Consider the following example  *end example*]
root (Root)	Specifies the top-most node of the diagram. [*Example*: Consider the following example 

Enumeration Value	Description
	end example]
self (Self)	Specifies the calling context node. [*Example*: Consider the following example  *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AxisType) is located in §A.5.3. *end note*]

21.4.7.7 ST_AxisTypes (Axis Type List)

This simple type represents a list of axis types.

This simple type allows a list of items of the ST_AxisType simple type (§21.4.7.6).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AxisTypes) is located in §A.5.3. *end note*]

21.4.7.8 ST_BendPoint (Bend Point)

This simple type defines where a bend is to occur within a connection between two nodes.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
beg (Beginning)	The bend is to occur at the beginning of the connection.
def (Default)	The default bend is used. By default connections bend in the center.
end (End)	The bend is to occur at the end of the connection.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BendPoint) is located in §A.5.3. *end note*]

21.4.7.9 ST_Booleans (Boolean List.)

A list of booleans.

This simple type allows a list of items of the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Booleans) is located in §A.5.3. *end note*]

21.4.7.10 ST_BoolOperator (Boolean Constraint)

This simple type specified Boolean operations which can be applied to compare constraints.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
equ (Equal)	Equal operator.
gte (Greater Than or Equal to)	Specifies the greater than or equal to Boolean operator.
lte (Less Than or Equal to)	Specifies the less than or equal to Boolean operator.
none (None)	Specifies a none Boolean operator

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BoolOperator) is located in §A.5.3. *end note*]

21.4.7.11 ST_Breakpoint (Breakpoint)

This simple type defines at what point the wrapping of nodes occurs for the snake algorithm.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bal (Balanced)	Specifies that the number of nodes in every row and every column should be equal.
endCnv (End of Canvas)	Specifies that nodes are added to the next column or row after filling the current column or row's space.
fixed (Fixed)	Specifies to use a user defined number of nodes in a column or row.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Breakpoint) is located in §A.5.3. *end note*]

21.4.7.12 ST_CenterShapeMapping (Center Shape Mapping)

This simple type defines the behavior of the cycle algorithm.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
fNode (First Node)	Specifies a node which is always in the center of a cycle diagram.
none (None)	Specifies the normal layout of a cycle diagram.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CenterShapeMapping) is located in §A.5.3. *end note*]

21.4.7.13 ST_ChildAlignment (Child Alignment)

This simple type defines how to align a node in its allocated space.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bottom)	Specifies to align the node to the bottom.
l (Left)	Specifies to align the node to the left.
r (Right)	Specifies to align the node to the right.
t (Top)	Specifies to align the node to the top.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ChildAlignment) is located in §A.5.3. *end note*]

21.4.7.14 ST_ChildDirection (Child Direction)

This simple type defines the layout direction of child nodes related to a specific parent node.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
horz (Horizontal)	Specifies that the child nodes are to be laid out

Enumeration Value	Description
	horizontally.
vert (Vertical)	Specifies that the child nodes are to be laid out vertically.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ChildDirection) is located in §A.5.3. *end note*]

21.4.7.15 ST_ChildOrderType (Child Order)

This simple type specifies the child order for a given layout node.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bottom)	Child order along the bottom.
t (Top)	Top child order.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ChildOrderType) is located in §A.5.3. *end note*]

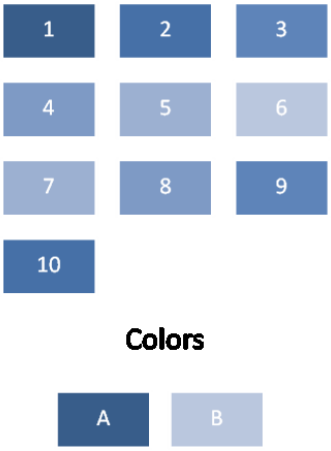
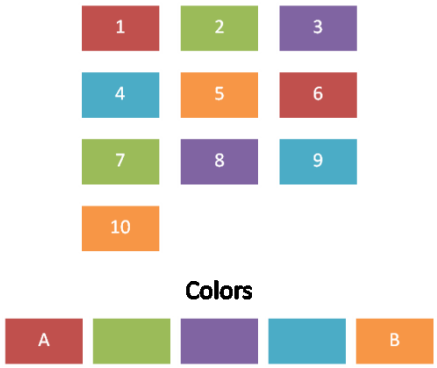
21.4.7.16 ST_ClrAppMethod (Color Application Method Type)

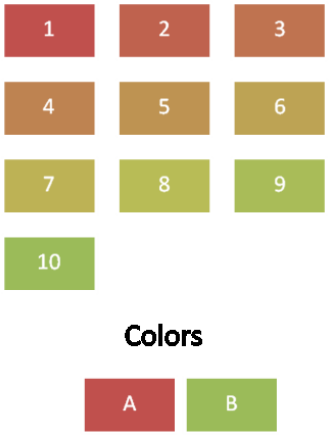
This simple type defines the way a given set of colors is applied to a set of nodes or items across a diagram.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
cycle (Cycle)	The colors apply from A to B to A if A and B were the colors present. [*Example*: Consider the following image as an example of cycle applied to a diagram:

Enumeration Value	Description
	 In this example, the color A is applied to node 1 and node 10. Color B is considered the node color between A and A across the diagram. Colors interpolate across the diagram from A to B back to A. *end example*]
repeat (Repeat)	The colors apply from A through B to A through B if A through B were the colors present. [*Example*: Consider the following image as an example of repeat applied to a diagram:  In this example, the color A is applied to node 1, the next color to node 2, and so on through color B, then this coloring is repeated until there are no more nodes to color. *end example*]
span (Span)	The colors interpolate from A to B across the entire diagram if A and B were the colors present. [*Example*: Consider the following image as an example of span applied to a diagram:

Enumeration Value	Description
	 In this example, the color A is applied to node 1, the color B is applied to node 10 and the colors applied to nodes 2 through 9 are interpolated between colors A and B. *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ClrAppMethod) is located in §A.5.3. *end note*]

21.4.7.17 ST_ConnectorDimension (Connector Dimension)

This simple type defines the dimensionality of the connection between two nodes.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
1D (1 Dimension)	Specifies a one dimensional connection, or rather a line.
2D (2 Dimensions)	Specifies a two dimensional connection which has both width and height.
cust (Custom)	Specifies a custom connection type.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ConnectorDimension) is located in §A.5.3. *end note*]

21.4.7.18 ST_ConnectorPoint (Connector Point)

This simple type defines different connection sites available on a node.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
auto (Auto)	Specifies that the algorithm determines the best connection site to use.
bCtr (Bottom Center)	Specifies that the bottom, center connection site is to be used.
bL (Bottom Left)	Specifies that the bottom, left connection site is to be used.
bR (Bottom Right)	Specifies that the bottom right connection site is to be used.
ctr (Center)	Specifies that the center connection site is to be used.
midL (Middle Left)	Specifies that the middle left connection site is to be used.
midR (Middle Right)	Specifies that the middle right connection site is to be used.
radial (Radial)	Specifies connections along a radial path to support the use of connections in cycle diagrams.
tCtr (Top Center)	Specifies that the top center connection site is to be used.
tL (Top Left)	Specifies that the top left connection site is to be used.
tR (Top Right)	Specifies that the top right connection site is to be used.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ConnectorPoint) is located in §A.5.3. *end note*]

21.4.7.19 ST_ConnectorRouting (Connector Routing)

This simple type defines how the routing of a connection between two nodes is supposed to progress from node 1 to node 2.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bend (Bending)	Specifies a bending connection which bends at a right angle.
curve (Curve)	Specifies a connection which is curved.
longCurve (Long Curve)	Specifies a connection that is curved that has a greater radius than a simple curved connection.
stra (Straight)	Specifies a straight connection.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ConnectorRouting) is located in §A.5.3. *end note*]

21.4.7.20 ST_ConstraintRelationship (Constraint Relationship)

This simple type specifies the types of constraint relationships which are present and can be used.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ch (Child)	The constraint should reference a child node.
des (Descendant)	The layout node can map to the descendants of the data point.
self (Self)	The layout node maps to the current data point.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ConstraintRelationship) is located in §A.5.3. *end note*]

21.4.7.21 ST_ConstraintType (Constraint Type)

This simple type defines the list of possible constraints available for use.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
alignOff (Alignment Offset)	This value defines the alignment offset for a node.
b (Bottom)	The bottom of the node.
begMarg (Beginning Margin)	Specifies the beginning margin.
begPad (Beginning Padding)	Specifies the beginning padding.
bendDist (Bending Distance)	Specifies the distance from the start of a connector to a bend in the connector.
bMarg (Bottom Margin)	Specifies the bottom margin.
bOff (Bottom Offset)	Specifies the bottom offset.
connDist (Connection Distance)	Specifies the connection distance.
ctrX (Center Height)	Specifies the center of the height.
ctrXOff (Center X Offset)	Specifies the center x coordinate offset.
ctrY (Center Width)	Specifies the center of the width.
ctrYOff (Center Y Offset)	Specifies the center y coordinate offset.

Enumeration Value	Description
diam (Diameter)	Specifies the diameter.
endMarg (End Margin)	Specifies the ending margin.
endPad (End Padding)	Specifies the end padding.
h (Height)	Specifies the height.
hArH (Arrowhead Height)	Specifies the height of the arrowhead portion of the connector.
hOff (Height Offset)	Specifies the amount to offset the height.
l (Left)	Specifies the left constraint.
lMarg (Left Margin)	Specifies the left margin.
lOff (Left Offset)	Specifies the left offset.
none (Unknown)	Unknown constraint.
primFontSz (Primary Font Size)	The primary font size.
pyraAcctRatio (Pyramid Accent Ratio)	Specifies the fraction of the width of the diagram that is reserved for the fly outs at their shortest distance.
r (Right)	Specifies the right constraint.
rMarg (Right Margin)	Specifies the right margin constraint.
rOff (Right Offset)	Specifies the right offset constraint.
secFontSz (Secondary Font Size)	The secondary font size.
secSibSp (Secondary Sibling Spacing)	The secondary sibling spacing.
sibSp (Sibling Spacing)	Specifies the minimum distance between sibling shapes.
sp (Spacing)	Specifies the spacing defined.
stemThick (Stem Thickness)	Specifies the thickness of the arrow's shaft.
t (Top)	Specifies the top constraint.
tMarg (Top Margin)	Top margin constraint.
tOff (Top Offset)	Top offset constraint.
userA (User Defined A)	User defined information.
userB (User Defined B)	User defined information.
userC (User Defined C)	User defined information.
userD (User Defined D)	User defined information.
userE (User Defined E)	User defined information.
userF (User Defined F)	User defined information.
userG (User Defined G)	User defined information.
userH (User Defined H)	User defined information.
userI (User Defined I)	User defined information.

Enumeration Value	Description
userJ (User Defined J)	User defined information.
userK (User Defined K)	User defined information.
userL (User Defined L)	User defined information.
userM (User Defined M)	User defined information.
userN (User Defined N)	User defined information.
userO (User Defined O)	User defined information.
userP (User Defined P)	User defined information.
userQ (User Defined Q)	User defined information.
userR (User Defined R)	User defined information.
userS (User Defined S)	User defined information.
userT (User Defined T)	User defined information.
userU (User Defined U)	User defined information.
userV (User Defined V)	User defined information.
userW (User Defined W)	User defined information.
userX (User Defined X)	User defined information.
userY (User Defined Y)	User defined information.
userZ (User Defined Z)	User defined information.
w (Width)	The width parameter.
wArH (Arrowhead Width)	Specifies the width of the arrowhead portion of the connector.
wOff (Width Offset)	Offsets the width by the specified amount.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ConstraintType) is located in §A.5.3. *end note*]

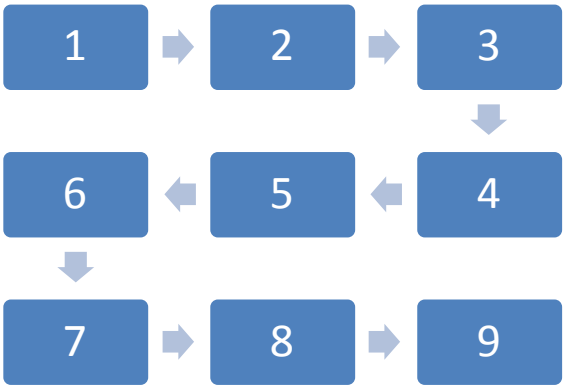
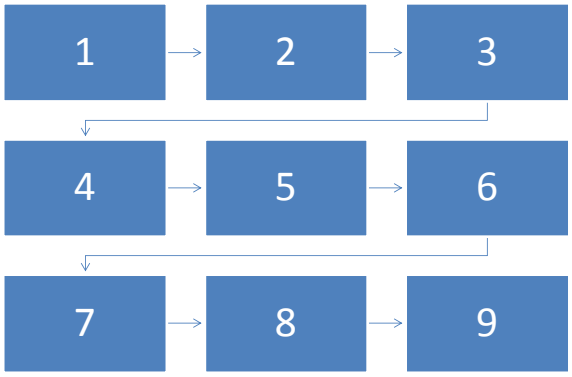
21.4.7.22 ST_ContinueDirection (Continue Direction)

This simple type specifies the behavior of the direction that additional nodes are added to new rows or columns in the snake algorithm.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
revDir (Reverse Direction)	Specifies that the direction is to be revered on a subsequent row or column. [*Example*: Consider the following diagram as an

Enumeration Value	Description
	example of reverse direction  *end example*]
sameDir (Same Direction)	Specifies that the direction is to be maintained on a subsequent row or column. [*Example*: Consider the following diagram as an example of same direction  *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ContinueDirection) is located in §A.5.3. *end note*]

21.4.7.23 ST_CxnType (Connection Type)

This simple type defines the different types of relationships that can be defined between two nodes.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
parOf (Parent Of)	This defines a parent-child relationship in the sense that node X is a parent of node Y.
presOf (Presentation Of)	A presentation type relationship. This type of relationship exists to actually present data.
presParOf (Presentation Parent Of)	A relationship defining a parent of a presentation node.
unknownRelationship (Unknown Relationship)	The type of relationship is unknown.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CxnType) is located in §A.5.3. *end note*]

21.4.7.24 ST_DiagramHorizontalAlignment (Horizontal Alignment)

This simple type defines the horizontal alignment.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ctr (Center)	Specifies center alignment.
l (Left)	Specifies left alignment.
none (None)	Specifies no alignment defined.
r (Right)	Specifies right alignment.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DiagramHorizontalAlignment) is located in §A.5.3. *end note*]

21.4.7.25 ST_DiagramTextAlignment (Text Alignment)

This simple type defines alignment types for text within a node.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ctr (Center)	Specifies center aligned text.
l (Left)	Specifies left aligned text.
r (Right)	Specifies right aligned text.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DiagramTextAlignment) is located in §A.5.3. *end note*]

21.4.7.26 ST_Direction (Diagram Direction Definition)

This simple type defines the possible values for a diagram's direction when displayed in an application.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
norm (Normal Direction)	This value specifies that the direction of the diagram should not be switched.
rev (Reversed Direction)	This value specifies that the direction of the diagram should be switched.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Direction) is located in §A.5.3. *end note*]

21.4.7.27 ST_ElementType (Data Point Type)

This simple type defines the different types of data points which are supported.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
all (All)	Defined as utilizing all of the nodes.
asst (Assistant)	The assistant nodes.
doc (Document)	Specifies the a node on the document level.
node (Node)	Data nodes that are children of other data nodes.
nonAsst (Non Assistant)	Selects all of the non-assistant nodes.
nonNorm (Non Normal)	Selects the non-normal elements.
norm (Normal)	Selects a normal elements.
parTrans (Parent Transition)	The transition associated with the parent node.
pres (Presentation)	This refers to a presentation node.
sibTrans (Sibling Transition)	Use only sibling transitions between data nodes. These transitions represent sibling relationships between nodes, and are frequently mapped to arrows between shapes in the drawing. A sibTrans value is sometimes used to create white space between nodes.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ElementType) is located in §A.5.3. *end note*]

21.4.7.28 ST_ElementTypes (Diagram Layout Node Type List)

A list of diagram layout node types.

This simple type allows a list of items of the ST_ElementType simple type (§21.4.7.27).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ElementTypes) is located in §A.5.3. *end note*]

21.4.7.29 ST_FallbackDimension (Fallback Dimension)

Specifies the dimensionality by which nodes can grow or shrink automatically.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
1D (1 Dimension)	Specifies that the node can grow or shrink by its height or its width, but not both.
2D (2 Dimensions)	Specifies that the node can grow or shrink by both height and width.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FallbackDimension) is located in §A.5.3. *end note*]

21.4.7.30 ST_FlowDirection (Flow Direction)

This simple type defines how the progression of new nodes are to be entered into the diagram.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
col (Column)	Specifies that the layout occurs in a column-based fashion. This would mean laying out the nodes from top to bottom, before moving left to right.
row (Row)	Specifies that the layout occurs in a row-based fashion. This would mean laying out the nodes from left to right before moving from top to bottom.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FlowDirection) is located in §A.5.3. *end note*]

21.4.7.31 ST_FunctionArgument (Function Argument)

Conditional expression function argument.

This simple type is a union of the following types:

- The ST_VariableType simple type (§21.4.7.64).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FunctionArgument) is located in §A.5.3. *end note*]

21.4.7.32 ST_FunctionOperator (Function Operator)

This simple type defines the condition expression functions which can be used to perform operations.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
equ (Equal)	Equal function operator.
gt (Greater Than)	Specifies the grater than function operator.
gte (Greater Than or Equal to)	Specifies the greater than or equal to function operator.
lt (Less Than)	Specifies the less than function operator.
lte (Less Than or Equal to)	Specifies the less than or equal to function operator.
neq (Not Equal To)	Specifies the not equal to function operator.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FunctionOperator) is located in §A.5.3. *end note*]

21.4.7.33 ST_FunctionType (Function Type)

This simple type defines the set of available conditional expression function types present for use.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
cnt (Count)	Specifies a count.
depth (Depth)	Specifies the depth.
maxDepth (Max Depth)	Defines the maximum depth.
pos (Position)	Retrieves the position of the node in the specified set of nodes.
posEven (Position Even)	Returns 1 if the specified node is at an even numbered

Enumeration Value	Description
	position in the data model.
posOdd (Position Odd)	Returns 1 if the specified node is in an odd position in the data model.
revPos (Reverse Position)	Reverse position function.
var (Variable)	Used to reference a variable.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FunctionType) is located in §A.5.3. *end note*]

21.4.7.34 ST_FunctionValue (Function Value)

Conditional expression function value.

This simple type is a union of the following types:

- The ST_AnimLvlStr simple type (§21.4.7.2).
- The ST_AnimOneStr simple type (§21.4.7.3).
- The ST_Direction simple type (§21.4.7.26).
- The ST_HierBranchStyle simple type (§21.4.7.37).
- The ST_ResizeHandlesStr simple type (§21.4.7.54).
- The W3C XML Schema boolean datatype.
- The W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FunctionValue) is located in §A.5.3. *end note*]

21.4.7.35 ST_GrowDirection (Grow Direction)

This simple type defines different starting locations for nodes within the snake algorithm.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bL (Bottom Left)	Specifies the placement of nodes is to start in the bottom left corner.
bR (Bottom Right)	Specifies the placement of nodes is to start in the bottom right corner.
tL (Top Left)	Specifies the placement of nodes is to start in the top left corner.
tR (Top Right)	Specifies the placement of nodes is to stat in the top right corner.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_GrowDirection) is located in §A.5.3. *end note*]

21.4.7.36 ST_HierarchyAlignment (Hierarchy Alignment)

This simple type defines different relative locations of child nodes and their descendants to a parent node within a hierarchy diagram.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bCtrCh (Bottom Center Child)	Specifies the child nodes are placed below the parent node and that they are center aligned to the parent node.
bCtrDes (Bottom Center Descendant)	Specifies the descendant nodes are placed below the parent node and that they are center aligned to the parent node.
bL (Bottom Left)	Specifies the child and descendant nodes are placed below the parent node and that the set is left aligned.
bR (Bottom Right)	Specifies the child and descendant nodes are placed below the parent node and the set is right aligned.
lB (Left Bottom)	Specifies the child and descendant nodes are placed to the left of the parent node and that the set is bottom aligned.
lCtrCh (Left Center Child)	Specifies the child nodes are placed to the left of the parent node and that the set is center aligned.
lCtrDes (Left Center Descendant)	Specifies the descendant nodes are placed to the left of the parent node and that the set is center aligned.
lT (Left Top)	Specifies the child and descendant nodes are placed to the left of the parent node and that the set is top aligned.
rB (Right Bottom)	Specifies the child and descendant nodes are placed to the right of the parent node and that the set is bottom aligned.
rCtrCh (Right Center Children)	Specifies the child nodes are placed to the right of the parent node and that the set is center aligned.
rCtrDes (Right Center Descendants)	Specifies the descendant nodes are placed to the right of the parent node and that the set is center aligned.
rT (Right Top)	Specifies the child and descendant nodes are placed to the right of the parent node and that the set is top

Enumeration Value	Description
	aligned.
tCtrCh (Top Center Children)	Specifies the child nodes are placed above the parent node and that the set is center aligned.
tCtrDes (Top Center Descendants)	Specifies the descendant nodes are placed above the parent node and that the set is center aligned.
tL (Top Left)	Specifies the child and descendant nodes are placed above the parent node and that the set is left aligned.
tR (Top Right)	Specifies the child and descendant nodes are placed above the parent node and that the set is right aligned.

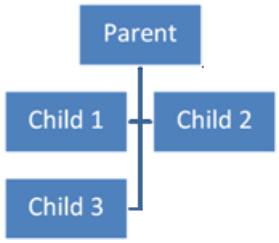
[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HierarchyAlignment) is located in §A.5.3. *end note*]

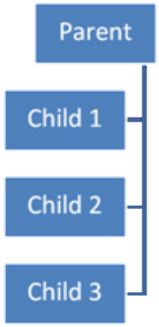
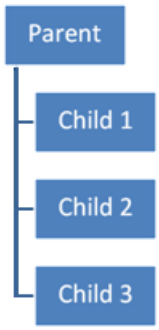
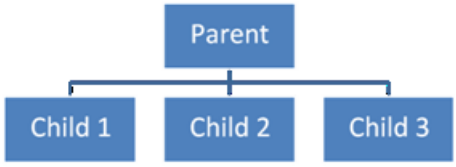
21.4.7.37 ST_HierBranchStyle (Hierarchy Branch Style Definition)

This simple type specifies the possible values for the branch style of a hierarchy diagram.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
hang (Hanging)	The branch style is hanging from the parent. [*Example*: Consider the following image as an example of a hanging branch style:  *end example*]
init (Initial)	This means that the value has not been set.
l (Left)	The branch style falls off the left. [*Example*: Consider the following image as an example of a left hanging branch style:

Enumeration Value	Description
	 *end example*]
r (Right)	The branch style falls off the right. [*Example*: Consider the following image as an example of a right hanging branch style:  *end example*]
std (Standard)	The standard branch style is to be used. [*Example*: Consider the following image as an example of a standard hanging branch style:  *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HierBranchStyle) is located in §A.5.3. *end note*]

21.4.7.38 ST_HueDir (Hue Direction)

When given two colors to interpolate between, one can go in one of two directions around the color wheel to perform the interpolation. This simple type defines that direction.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ccw (Counterclockwise Hue Direction)	A hue interpolation in the counterclockwise direction.
cw (Clockwise Hue Direction)	A hue interpolation in the clockwise direction.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HueDir) is located in §A.5.3. *end note*]

21.4.7.39 ST_Index1 (1-Based Index)

A 1-based index.

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 1.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Index1) is located in §A.5.3. *end note*]

21.4.7.40 ST_Ints (Integer List)

A list of integers.

This simple type allows a list of items of the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Ints) is located in §A.5.3. *end note*]

21.4.7.41 ST_LayoutShapeType (Layout Shape Type)

All of the available shape types.

This simple type is a union of the following types:

- The ST_OutputShapeType simple type (§21.4.7.48).
- The ST_ShapeType simple type (§20.1.10.55).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LayoutShapeType) is located in §A.5.3. *end note*]

21.4.7.42 ST_LinearDirection (Linear Direction)

This simple type defines the direction of growth of new nodes.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
fromB (From Bottom)	Specifies growth to start from the bottom.
fromL (From Left)	Specifies growth to start from the left.
fromR (From Right)	Specifies growth to start from the right.
fromT (From Top)	Specifies growth to start from the Top

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LinearDirection) is located in §A.5.3. *end note*]

21.4.7.43 ST_ModelId (Model Identifier)

The unique ID of the element within the data model. Model Identifiers can be either longs or guids.

This simple type is a union of the following types:

- The ST_Guid simple type (§22.9.2.4).
- The W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ModelId) is located in §A.5.3. *end note*]

21.4.7.44 ST_NodeCount (Number of Nodes Definition)

This simple type defines a count of the number of nodes for a property in a diagram. A value of -1 shall mean that the value is unbounded.

This simple type's contents are a restriction of the W3C XML Schema int datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to -1.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_NodeCount) is located in §A.5.3. *end note*]

21.4.7.45 ST_NodeHorizontalAlignment (Node Horizontal Alignment)

This simple type defines the horizontal alignment of a node.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ctr (Center)	Specifies center alignment.
l (Left)	Specifies left alignment.
r (Right)	Specifies right alignment.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_NodeHorizontalAlignment) is located in §A.5.3. *end note*]

21.4.7.46 ST_NodeVerticalAlignment (Node Vertical Alignment)

This simple type defines the vertical alignment of a node.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bottom)	Specifies bottom alignment.
mid (Middle)	Specifies middle alignment.
t (Top)	Specifies top alignment.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_NodeVerticalAlignment) is located in §A.5.3. *end note*]

21.4.7.47 ST_Offset (Offset)

This simple type defines whether or not subsequent rows or columns in the snake algorithm are offset from the preceding row or column.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ctr (Center)	Specifies no offset.
off (Offset)	Specifies that the nodes are shifted by some amount relative to the preceding row or column.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Offset) is located in §A.5.3. *end note*]

21.4.7.48 ST_OutputShapeType (Output Shape Type)

Shapes which are special specifically for a DrawingML diagram.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
conn (Connection)	Connection shape type.
none (None)	None.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_OutputShapeType) is located in §A.5.3. *end note*]

21.4.7.49 ST_ParameterId (Parameter Identifier)

This simple type defines algorithm parameters which can be modified in order to adjust the behavior of algorithms for use in layout nodes.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
alignTx (Text Alignment)	This value defines how the text is aligned in a node.
ar (Aspect Ratio)	Specifies the aspect ratio (width to height) of the composite node to use when determining child constraints. A value of 0 means leave the width and height constraints as is. The algorithm can temporarily shrink one dimension to achieve that ratio. [*Example*: If a composite node has a width constraint of 20 and height constraint of 10, and if ar=1.5, composite uses a width value of 15 to calculate the composite node's child constraints. However, the algorithm does not propagate this value to other nodes. *end example*]
autoTxRot (Auto Text Rotation)	Auto text rotation.
begPts (Beginning Points)	Beginning Points
begSty (Beginning Arrowhead Style)	Beginning Arrowhead Style
bendPt (Bend Point)	The bend point.
bkpt (Breakpoint)	Specifies the point at which the diagram starts to snake. The value bal specifies that snaking begin at an

Enumeration Value	Description
	even number of rows and columns. The value fixed specifies that snaking begin at a fixed point. The value endCnv specifies that snaking begin when there is no more room for a shape in the row.
bkPtFixedVal (Breakpoint Fixed Value)	Specifies where the snake should break, if bkpt=fixed.
chAlign (Child Alignment)	Specifies the alignment of the children.
chDir (Child Direction)	The child direction.
connRout (Connection Route)	The route of the connection.
contDir (Continue Direction)	Specifies the direction of the subsequent row or column. [*Example*: If the algorithm initially places the nodes from left to right, revDir places the nodes in the next row from right to left. However if the algorithm uses contDir, the nodes on the next row are arranged from left to right. *end example*]
ctrShpMap (Center Shape Mapping)	Specifies where to place nodes in relation to the center circle.
dim (Connector Dimension)	Specifies the connector dimension.
dstNode (Destination Node)	Specifies the name of the layout node from which to end the connection from.
endPts (End Points)	Specifies the end points.
endSty (End Style)	Specifies the end style.
fallback (Fallback Scale)	1D specifies fallback. It only scales in one dimension. 2D specifies fallback. It scales in both dimensions equally.
flowDir (Flow Direction)	Specifies whether nodes are arranged in rows or columns.
grDir (Grow Direction)	Specifies from which corner the snake grows. [*Example*: If the algorithm uses a top left value, the snake grows from the top left. *end example*]
hierAlign (Hierarchy Alignment)	The alignment of the hierarchy.
horzAlign (Horizontal Alignment)	Aligns all the child nodes within the space reserved for the parent and adjusts child positions in the x direction.
linDir (Linear Direction)	Specifies the linear direction.
lnSpAfChP (Line Spacing After Children Paragraph)	Line spacing after children.
lnSpAfParP (Line Spacing After Parent Paragraph)	Line spacing after the parent.
lnSpCh (Line Spacing Children)	Line spacing of the children
lnSpPar (Line Spacing Parent)	Line spacing of the parent.

Enumeration Value	Description
nodeHorzAlign (Node Horizontal Alignment)	Specifies how child nodes are aligned within the extents of the canvas. [*Example*: You can align the tops of all the child nodes, but center all of them within the canvas. *end example*]
nodeVertAlign (Node Vertical Alignment)	Specifies how child nodes are aligned within the extents of the canvas. Same as nodeHorzAlign, but in the y direction.
off (Offset)	Specifies the offset.
parTxLTRAlign (Parent Text Left-to-Right Alignment)	Specifies the paragraph alignment of parent text when the shape has only parent text. This parameter applies when the text direction is left to right.
parTxRTLAlign (Parent Text Right-to-Left Alignment)	Specifies the paragraph alignment of parent text when the shape has only parent text. This parameter applies when the text direction is right to left.
pyraAcctBkgdNode (Pyramid Accent Background Node)	If pyramid has a composite child node, specifies the name of the node that is a child of the composite that makes up the child flyout shape. If the node specifies a shape of the nonIsoscelesTrapezoid autoshape, it modifies the adjust handles in order to fit the flyout flush against the side of the pyramid.
pyraAcctPos (Pyramid Accent Position)	Specifies the placement of the flyout grandchildren.
pyraAcctTxMar (Pyramid Accent Text Margin)	Specifies the placement of one edge of the child text (grandchild node). If the value is step, the text is against the edge of the pyramid. If the value is stack, the text aligns.
pyraAcctTxNode (Pyramid Accent Text Node)	If pyramid has a composite child node, specifies the child node that should hold the child text.
pyraLvlNode (Pyramid Level Node)	If pyramid has a composite child node, specifies the name of the node that is a child of the composite that makes up the pyramid itself. If the node specifies a trapezoid shape, it modifies the adjustment handles to construct a pyramid.
rotPath (Rotation Path)	The rotation path specified.
rtShortDist (Route Shortest Distance)	If true, the connector is routed through the shortest distance between the points.
secChAlign (Secondary Child Alignment)	The secondary child alignment.
secLinDir (Secondary Linear Direction)	The secondary linear direction.
shpTxLTRAlignCh (Shape Text Left-to-Right Alignment)	Specifies the paragraph alignment of all text within the shape when the shape contains both parent and child text. This parameter applies when the text direction is left to right.

Enumeration Value	Description
shpTxRTLAlignCh (Shape Text Right-to-Left Alignment)	Specifies the paragraph alignment of all text within the shape when the shape contains both parent and child text. This parameter applies when the text direction is right to left.
spanAng (Span Angle)	Specifies the angle the cycle spans. Final shapealign text is placed at stAng+spanAng, unless spanAng=360. In that case, the algorithm places the text so that shapes do not overlap.
srcNode (Source Node)	Specifies the name of the layout node from which to start the connection.
stAng (Start Angle)	Specifies the angle at which the first shape is placed. Angles are in degrees, measured clockwise from a line pointing straight upward from the center of the cycle.
stBulletLvl (Start Bullets At Level)	Specifies whether bullets start at the top level (1) or with children (2).
stElem (Start Element)	Specifies the point type of the layout node to use as the first shape in the cycle.
txAnchorHorz (Text Anchor Horizontal)	Specifies the y-axis position of the text area within a shape.
txAnchorHorzCh (Text Anchor Horizontal With Children)	Specifies that the definition can allow a different text anchoring on the x-axis, if child nodes exist in the shape.
txAnchorVert (Text Anchor Vertical)	Specifies the x-axis position of the text area within a shape.
txAnchorVertCh (Text Anchor Vertical With Children)	Specifies that the definition can allow a different text anchoring on the y-axis, if child nodes exist in the shape.
txBlDir (Text Block Direction)	Specifies whether the text block is vertical or horizontal.
txDir (Text Direction)	Specifies where the text of the first node starts.
vertAlign (Vertical Alignment)	Aligns all the child nodes within the space reserved for the parent and adjusts child positions in the y direction.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ParameterId) is located in §A.5.3. *end note*]

21.4.7.50 ST_ParameterVal (Parameter Values)

Specifies the list of parameter types that can be used by a diagram.

This simple type is a union of the following types:

- The ST_ArrowheadStyle simple type (§21.4.7.4).
- The ST_AutoTextRotation simple type (§21.4.7.5).
- The ST_BendPoint simple type (§21.4.7.8).
- The ST_Breakpoint simple type (§21.4.7.11).
- The ST_CenterShapeMapping simple type (§21.4.7.12).
- The ST_ChildAlignment simple type (§21.4.7.13).
- The ST_ChildDirection simple type (§21.4.7.14).
- The ST_ConnectorDimension simple type (§21.4.7.17).
- The ST_ConnectorPoint simple type (§21.4.7.18).
- The ST_ConnectorRouting simple type (§21.4.7.19).
- The ST_ContinueDirection simple type (§21.4.7.22).
- The ST_DiagramHorizontalAlignment simple type (§21.4.7.24).
- The ST_DiagramTextAlignment simple type (§21.4.7.25).
- The ST_FallbackDimension simple type (§21.4.7.29).
- The ST_FlowDirection simple type (§21.4.7.30).
- The ST_GrowDirection simple type (§21.4.7.35).
- The ST_HierarchyAlignment simple type (§21.4.7.36).
- The ST_LinearDirection simple type (§21.4.7.42).
- The ST_NodeHorizontalAlignment simple type (§21.4.7.45).
- The ST_NodeVerticalAlignment simple type (§21.4.7.46).
- The ST_Offset simple type (§21.4.7.47).
- The ST_PyramidAccentPosition simple type (§21.4.7.52).
- The ST_PyramidAccentTextMargin simple type (§21.4.7.53).
- The ST_RotationPath simple type (§21.4.7.55).
- The ST_SecondaryChildAlignment simple type (§21.4.7.56).
- The ST_SecondaryLinearDirection simple type (§21.4.7.57).
- The ST_StartingElement simple type (§21.4.7.58).
- The ST_TextAnchorHorizontal simple type (§21.4.7.59).
- The ST_TextAnchorVertical simple type (§21.4.7.60).
- The ST_TextBlockDirection simple type (§21.4.7.61).
- The ST_TextDirection simple type (§21.4.7.62).
- The ST_VerticalAlignment simple type (§21.4.7.65).
- The W3C XML Schema boolean datatype.
- The W3C XML Schema double datatype.
- The W3C XML Schema int datatype.
- The W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ParameterVal) is located in §A.5.3. *end note*]

21.4.7.51 ST_PtType (Point Type)

This simple type defines the different point types which can be utilized to create diagrams in DiagramML.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
asst (Assistant Element)	This point type is used in a hierarchy diagram to represent an assistant element.
doc (Document)	This point type specifies a document type point. This point type can be thought of as the root node associated with the document itself.
node (Node)	The node point type specifies a basic point type.
parTrans (Parent Transition)	This point type specifies a parent transition element.
pres (Presentation)	Specifies a presentation point type.
sibTrans (Sibling Transition)	This point type specifies a sibling transition element.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PtType) is located in §A.5.3. *end note*]

21.4.7.52 ST_PyramidAccentPosition (Pyramid Accent Position)

This simple type defines different positioning for the accent shapes which can be associated with a pyramid algorithm.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
aft (Pyramid Accent After)	Specifies that the accent shapes are to be placed to the right of the pyramid.
bef (Before)	Specifies that the accent shapes are to be placed to the left of the pyramid.

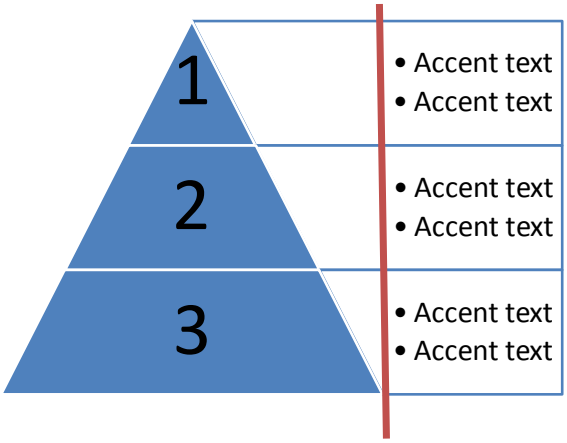
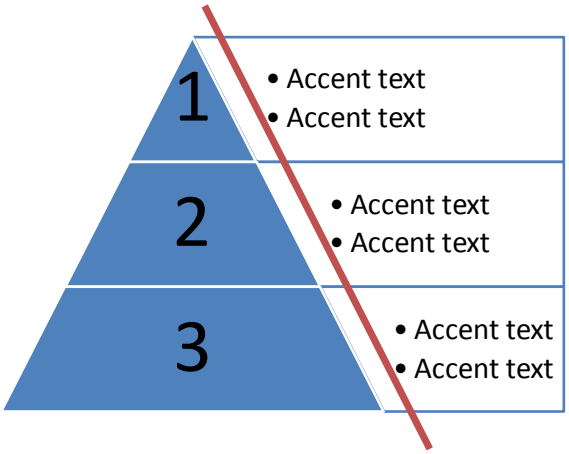
[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PyramidAccentPosition) is located in §A.5.3. *end note*]

21.4.7.53 ST_PyramidAccentTextMargin (Pyramid Accent Text Margin)

This simple type defines different ways to lay out text in the accent shape for a pyramid algorithm.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
stack (Stack)	Specifies that all accent shape text is to be left aligned. [*Example*: Consider the following example of a pyramid diagram  *end example*]
step (Step)	Specifies that all accent shape text is to be relative to the pyramid. [*Example*: Consider the following example of a pyramid diagram  *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PyramidAccentTextMargin) is located in §A.5.3. *end note*]

21.4.7.54 ST_ResizeHandlesStr (Resize Handle)

This simple type defines the possible behaviors when resizing shapes within a diagram. Because the size of the shape plays a large role in the overall layout of other nodes within the diagram, there are two ways resize can occur on a node.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
exact (Exact)	This value specifies that the resize of the shape occurs and sizes exactly to the size the user defines, which causes all other shapes in the diagram to shrink or grow accordingly.
rel (Relative)	This value specifies that resize operations happen relatively. This means that the relative size difference between nodes is maintained before and after the resize operation.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ResizeHandlesStr) is located in §A.5.3. *end note*]

21.4.7.55 ST_RotationPath (Rotation Path)

This simple type defines rotation properties for nodes within the cycle algorithm.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
alongPath (Along Path)	Specifies that the nodes should rotate in relation to their placement along the cycle.
none (None)	Specifies that the nodes should not rotate.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RotationPath) is located in §A.5.3. *end note*]

21.4.7.56 ST_SecondaryChildAlignment (Secondary Child Alignment)

This simple type defines different alignment properties of the both hanging layout type of the hierarchy algorithm.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bottom)	Specifies that the children nodes should be bottom aligned.
l (Left)	Specifies that the children nodes should be left aligned.
none (None)	Specifies no alignment.
r (Right)	Specifies that the children nodes should be right aligned.
t (Top)	Specifies that the children nodes should be top aligned.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SecondaryChildAlignment) is located in §A.5.3. *end note*]

21.4.7.57 ST_SecondaryLinearDirection (Secondary Linear Direction)

This simple type defines different directions for the nodes in a both hanging layout in the hierarchy algorithm.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
fromB (From Bottom)	Specifies that the nodes begin from the bottom and move upward.
fromL (From Left)	Specifies that the nodes begin from the left and move right.
fromR (From Right)	Specifies that the nodes begin from the right and move left.
fromT (From Top)	Specifies that the nodes begin from the top and move downward.
none (None)	Specifies no direction.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SecondaryLinearDirection) is located in §A.5.3. *end note*]

21.4.7.58 ST_StartingElement (Starting Element)

This simple type defines behavior for the first node in a cycle algorithm.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
node (Node)	Specifies that a node should be placed first.
trans (Transition)	Specifies that a transition should be placed first.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_StartingElement) is located in §A.5.3. *end note*]

21.4.7.59 ST_TextAnchorHorizontal (Text Anchor Horizontal)

This simple type defines horizontal anchor points for text.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ctr (Center)	Specifies text to be anchored to the center.
none (None)	Specifies no horizontal text anchor.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextAnchorHorizontal) is located in §A.5.3. *end note*]

21.4.7.60 ST_TextAnchorVertical (Text Anchor Vertical)

This simple type defines vertical anchor points for text.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bottom)	Specifies text to be anchored to the bottom.
mid (Middle)	Specifies text to be anchored to the middle.
t (Top)	Specifies text to be anchored to the top.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextAnchorVertical) is located in §A.5.3. *end note*]

21.4.7.61 ST_TextBlockDirection (Text Block Direction)

This simple type defines different layout options for text within a node.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
horz (Horizontal)	Specifies that the text is to be horizontal.
vert (Vertical Direction)	Specifies that the text is to be vertical.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextBlockDirection) is located in §A.5.3. *end note*]

21.4.7.62 ST_TextDirection (Text Direction)

This simple type defines different way the growth of additional text can occur within a node.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
fromB (From Bottom)	Specifies additional text grows from the bottom.
fromT (From Top)	Specifies additional text grows from the top.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextDirection) is located in §A.5.3. *end note*]

21.4.7.63 ST_UnsignedInts (Unsigned Integer List)

A list of unsigned integers.

This simple type allows a list of items of the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_UnsignedInts) is located in §A.5.3. *end note*]

21.4.7.64 ST_VariableType (Variable Type)

Conditional expression variable type.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
animLvl (Animation Level)	Specifies the animation level
animOne (Animate One)	Specifies animate as one.
bulEnabled (Bullets Enabled)	Specifies bullets enabled.
chMax (Child Max)	The maximum number of children.

Enumeration Value	Description
chPref (Child Preference)	The preferred number of children.
dir (Direction)	Specifies the direction of the diagram.
hierBranch (Hierarchy Branch)	The hierarchy branch.
none (Unknown)	Unknown variable type.
orgChart (Organizational Chart Algorithm)	Algorithm that lays out an org chart.
resizeHandles (Resize Handles)	Specifies the resize handles.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_VariableType) is located in §A.5.3. *end note*]

21.4.7.65 ST_VerticalAlignment (Vertical Alignment)

This simple type defines different vertical alignment parameters.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bottom)	Specifies bottom aligned.
mid (Middle)	Specifies middle aligned.
none (None)	Specifies no vertical alignment.
t (Top)	Specifies top aligned.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_VerticalAlignment) is located in §A.5.3. *end note*]

21.4.7.66 ST_PrSetCustVal (Property Set Customized Value)

This simple type defines customization percentage values for certain elements in DrawingML.

This simple type is a union of the following types:

The ST_Percentage simple type (§22.9.2.9).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PrSetCustVal) is located in §A.5.3. *end note*]

22. Shared MLs Reference Material

[No documentation has been entered for this section heading.]

22.1 Math

The following documentation specifies the XML representation of mathematical text for OOXML. This shared ML is known as that Office Math Markup Language (OMML). Mathematical text represented by OMML includes but is not limited to: equations, expressions, formulas, matrices, and other mathematical elements. The outermost OMML element of an instance of mathematical text in display mode is oMathPara, a math paragraph of one or more instances of mathematical text. Each instance of mathematical text inside the math paragraph is represented as a single oMath. Inside each oMath is a combination of mathematical runs (r) and objects or functions such as accents (acc) or fractions (f).

22.1.1 Table of Contents

This subclause is informative.

End of informative text.

22.1.2 Elements

The following elements describe the contents of mathematical text.

22.1.2.1 acc (Accent)

This element specifies the accent function, consisting of a base and a combining diacritical mark. If accPr is omitted, the default accent is U+0302 (COMBINING CIRCUMFLEX ACCENT). [*Example*: Example accent functions are $\dot{a}$, $\acute{a}$, and $\tilde{a}$.

```
<m:acc>
  <m:accPr>
    <m:chr m:val="&#771;"/>
    <m:ctrlPr/>
  </m:accPr>
  <m:e>
    <m:r>
      <m:t>a</m:t>
    </m:r>
  </m:e>
</m:acc>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Acc) is located in §A.6.1. *end note*]

22.1.2.2 accPr (Accent Properties)

This element specifies the properties of the Accent function. If chr is omitted, the default accent character is U+0302 (COMBINING CIRCUMFLEX ACCENT). [*Example*:

The diacritical mark ~ (tilde) is:

```
<m:accPr>
  <m:chr m:val="&#771;"/>
</m:accPr>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_AccPr) is located in §A.6.1. *end note*]

22.1.2.3 aln (Alignment)

This element specifies the alignment property on the box object. It is utilized only when the box is designated as an operator emulator. When 1 or true, this operator emulator serves as an alignment point; that is, designated alignment points in other equations can be aligned with it.

When the element is absent, the default is for the parent structure to not be an alignment point.

When the element is present and the val attribute is absent, the default value of the val attribute is 1 meaning that this property's parent structure is an alignment point.

[*Example*: For example, the following equation uses the operator emulator as an alignment point: $a == b$.

Its XML representation is as follows:

```
<m:oMath>
  <m:r>
    <m:t>a</m:t>
  </m:r>
  <m:box>
    <m:boxPr>
      <m:opEmu m:val="1"/>
      <m:aln m:val="1"/>
      <m:ctrlPr/>
    </m:boxPr>
    <m:e>
      <m:r>
        <m:t>==</m:t>
      </m:r>
    </m:e>
  </m:box>
```

```
    <m:r>
      <m:t>b</m:t>
    </m:r>
  </m:oMath>
```

end example]

[*Example*:

Given the following equations:

$$a + b - c = 2$$
$$x - y + z = 3$$

Their associated XML represenation is as follows:

```
  <m:oMathPara>
    <m:oMath>
      <m:r>
        <m:t>a+b</m:t>
      </m:r>
      <m:r>
        <m:rPr>
          <m:aln/>
        </m:rPr>
        <m:t>-c=2</m:t>
      </m:r>
      <m:r>
        <m:rPr>
          <m:sty m:val="p"/>
        </m:rPr>
        <w:br/>
      </m:r>
    </m:oMath>
    <m:oMath>
      <m:r>
        <m:t>x</m:t>
      </m:r>
      <m:r>
        <m:rPr>
          <m:aln/>
        </m:rPr>
        <m:t>-y+z=3</m:t>
      </m:r>
    </m:oMath>
```



```
</m:oMathPara>
```

The subtraction symbols in each of the above equations have been identified as alignment points, so the equations are aligned at their subtraction symbols (because they are in the same Math paragraph).

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.4 alnScr (Align Scripts)

This element specifies the alignment of scripts in the subscript/superscript function. When `1` or `true`, subscripts and superscripts are aligned to each other. When `0` or `false`, they are kerned to the shape of the base. If this element is omitted, scripts are not aligned. In other words, when the element is absent, the default is for the sub-superscript object to not align the superscript and subscript with each other.

When the element is present and the val attribute is absent, the default value of the val attribute is 1 meaning that this property is applied.

[*Example*: Example (OFF): f_2^3; Example (ON): f_2^3.

The XML representation of the second example above is:

```
<m:sSubSup>
  <m:sSubSupPr>
    <m:alnScr   m:val="1"/>
  </m:sSubSupPr>
  <m:e>
    <m:r>
      <m:t>f</m:t>
    </m:r>
  </m:e>
```

```
    <m:sub>
      <m:r>
        <m:rPr>
          <m:scr m:val="roman"/>
          <m:sty m:val="p"/>
        </m:rPr>
        <m:t>2</m:t>
      </m:r>
    </m:sub>
    <m:sup>
      <m:r>
        <m:rPr>
          <m:scr m:val="roman"/>
          <m:sty m:val="p"/>
        </m:rPr>
        <m:t>3</m:t>
      </m:r>
    </m:sup>
  </m:sSubSup>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.5 argPr (Argument Properties)

This element specifies any properties of the math argument. [*Example*: The XML below represents the argSz attribute on the base element e of a box:

```
<m:box>
  <m:boxPr>
    <m:noBreak m:val="0"/>
    <m:ctrlPr/>
  </m:boxPr>
  <m:e>
    <m:argPr>
      <m:argSz m:val="-1"/>
    </m:argPr>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:box>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathArgPr) is located in §A.6.1. *end note*]

22.1.2.6 argSz (Argument Size)

This element specifies the size, or script level, of an argument. If the element is omitted, the default argument size is 0.

Whether the element is absent or present without the val attribute, the default value of the val attribute is always `0`. However, this property does not specify an absolute size, rather it's absolute to it's starting size. Most arguments begin at the normal size which will be defined as `0` in the list below, but some begin, by default smaller or larger than the normal size. For instance, `1` represents that it is one size larger than normal, `2` represents that it is two sizes larger than normal, `-1` represents that it is one size smaller than normal, and `-2` represents that it is two sizes smaller than normal. The list below indicates the default starting sizes for each argument, if it is not in this list, then the size cannot be changed on that argument.

- box base: 0
- group-character base: 0
- lower limit limit: -1
- upper limit limit: -1
- n-ary operator subscript: -1
- n-ary operator superscript: -1
- radical degree: -2
- pre-sub-superscript subscript: -1
- pre-sub-superscript superscript: -1
- subscript subscript: -1
- sub-supersript subscript: -1

- sub-supersript superscript: -1
- superscript superscript: -1

[*Example*: The following example contains three runs: regular mathematical text, a box object with the base at script size (val=-1) and a box object with the base at script-script size (val=-2).

$$abc^{abc\ abc}$$

The XML below shows argSz used in the middle box.

```
<m:box>
  <m:boxPr>
    <m:noBreak m:val="0"/>
  </m:boxPr>
  <m:e>
    <m:argPr>
      <m:argSz m:val="-1"/>
    </m:argPr>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:box>
```

Because the size is set to -1 on the degree argument, the degree argument is 1 size smaller normal.

end example]

Attributes	Description
val (Value)	Specifies a value between -2 and 2 for the property defined by the parent XML element. The positive or negative sign specifies in which direction to change argument size; the absolute value specifies by how much. The table below represents two cases in which argument size can be changed: superscripts and boxes. In the superscript object a^{b^c}, by default the term c has script-script size. Should the user wish for the c to be shown at script size, `val` should be set to +1 (that is, one size larger). Should the user wish for c to be shown at text size, `val` should be set to +2 (that is, two sizes larger).

`val` of c in a^{b^c}	Display
Default	a^{b^c}
+1	a^{b^c}
+2	a^{b^c}

Attributes	Description
	<table><tr><th>val of abc</th><th>Display</th></tr><tr><td>Default</td><td>abc</td></tr><tr><td>-1</td><td>abc</td></tr><tr><td>-2</td><td>abc</td></tr></table> The possible values for this attribute are defined by the ST_Integer2 simple type (§22.1.3.5).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Integer2) is located in §A.6.1. *end note*]

22.1.2.7 bar (Bar)

This element specifies the bar function, consisting of a base argument and an overbar or underbar, as in $\overline{a}$ and $\underline{a}$.

[*Example*: The XML below demonstrates the overbar in use.

```
<m:bar>
  <m:barPr>
   <m:pos   m:val="top"/>
  </m:barPr>
  <m:e>
    <m:r>
      <m:t>a</m:t>
    </m:r>
  </m:e>
</m:bar>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Bar) is located in §A.6.1. *end note*]

22.1.2.8 barPr (Bar Properties)

This element specifies properties of the bar function. If this element is omitted, the bar assumes its default location of top (the mathematical overbar). [*Example*:

$\underline{x}$ (x with an underbar) is represented by the following XML representation:

```
<m:bar>
  <m:barPr>
    <m:pos m:val="bot"/>
```

```
    </m:barPr>
    <m:e>
      <m:r>
        <m:t>x</m:t>
      </m:r>
    </m:e>
  </m:bar>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_BarPr) is located in §A.6.1. *end note*]

22.1.2.9 baseJc (Matrix Base Justification)

This element specifies the justification of the matrix. Text outside of the matrix can be aligned with the bottom, top, or center of a matrix object. If this element is omitted, the matrix assumes center justification. In other words, whether the element is absent or present without the val attribute, the default of the val attribute is `center`.

[*Example*: This matrix has center baseJc: $\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$

This matrix has top baseJc: $\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$

This matrix has bottom baseJc: $\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$

The XML below represents the matrix with top baseJC:

```
<m:d>
  <m:dPr>
    <m:shp  m:val="match"/>
  </m:dPr>
  <m:e>
    <m:m>
      <m:mPr>
        <m:baseJc m:val="top"/>
        <m:mcs>
          <m:mc>
            <m:mcPr>

              <m:count m:val="2"/>
```



```
        <m:mcJc  m:val="center"/>
      </m:mcPr>
    </m:mc>
  </m:mcs>
</m:mPr>
<m:mr>
  <m:e>
    <m:r>
      <m:rPr>
        <m:scr  m:val="roman"/>
        <m:sty m:val="p"/>
      </m:rPr>
      <m:t>1</m:t>
    </m:r>
  </m:e>
  <m:e>
    <m:r>
      <m:rPr>
        <m:scr  m:val="roman"/>
        <m:sty  m:val="p"/>
      </m:rPr>
      <m:t>2</m:t>
    </m:r>
  </m:e>
</m:mr>
<m:mr>
  <m:e>
    <m:r>
      <m:rPr>
        <m:scr  m:val="roman"/>
        <m:sty  m:val="p"/>
      </m:rPr>
      <m:t>3</m:t>
    </m:r>
  </m:e>
```

```
          <m:e>
            <m:r>
              <m:rPr>
                <m:scr  m:val="roman"/>
                <m:sty  m:val="p"/>
              </m:rPr>
              <m:t>4</m:t>
            </m:r>
          </m:e>
        </m:mr>
        <m:mr>
          <m:e>
            <m:r>
              <m:rPr>
                <m:scr  m:val="roman"/>
                <m:sty  m:val="p"/>
              </m:rPr>
              <m:t>5</m:t>
            </m:r>
          </m:e>
          <m:e>
            <m:r>
              <m:rPr>
                <m:scr  m:val="roman"/>
                <m:sty  m:val="p"/>
              </m:rPr>
              <m:t>6</m:t>
            </m:r>
          </m:e>
        </m:mr>
      </m:m>
    </m:e>
  </m:d>
```

Because the matrix base justification is top, the top row of the matrix is lined up with the baseline of the rest of the line.

end example]

Attributes	Description
val (Value)	Specifies the vertical justification parent element respect to surrounding text. Possible values are `top`, `bottom`, and `center`. [*Example*: The following examples illustrate baseJc on the matrix object m.

Attributes	Description
	This matrix has center baseJc: $\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$ This matrix has top baseJc: $\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$ This matrix has bottom baseJc: $\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$ The possible values for this attribute are defined by the ST_YAlign simple type (§22.9.2.20).

[*Note*: The W3C XML Schema definition of this element's content model (CT_YAlign) is located in §A.6.1. *end note*]

22.1.2.10 begChr (Delimiter Beginning Character)

This element specifies the beginning, or opening, delimiter character. Mathematical delimiters are enclosing characters such as parentheses, brackets, and braces. If this element is omitted, the default begChr is '('. In other words, when the element is absent, the default is for the delimiter object beginning character to be Unicode character U+0028 (LEFT PARENTHESIS).

When the element is present and the val attribute is absent, the default of the val attribute is empty which means the delimiter object beginning character is absent.

[*Example*: In the following example, $\{a\}$ uses { and } as its enclosing characters:

```
<m:dPr>
  <m:begChr  m:val="{"/>
  <m:endChr m:val="}"/>
</m:dPr>
```

end example]

Attributes	Description

val (value)	Specifies the character used by the parent element. When it is omitted, the parent uses its assigned default.
	
	[*Example*: Accent object ã:
	
	`<m:accPr>`
	`  <m:chr m:val="̃"/>`
	`</m:accPr>`
	
	end example]

Attributes	Description
	The possible values for this attribute are defined by the ST_Char simple type (§22.1.3.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Char) is located in §A.6.1. *end note*]

22.1.2.11 borderBox (Border-Box Object)

This element specifies the Border Box object, consisting of a border drawn around an instance of mathematical text (such as a formula or equation), as in $\boxed{a^2 + b^2 = c^2}$. If borderBoxPr is omitted then the default behavior of borderBox is a rectangular border (as shown in the "abc" example below).

[*Example*: The following example shows the XML representation of the following Border Box: $\boxed{abc}$

```
<m:borderBox>
  <m:e>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:borderBox>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_BorderBox) is located in §A.6.1. *end note*]

22.1.2.12 borderBoxPr (Border-Box Properties)

This element specifies the properties of the Border Box object, which dictate the types of lines that can be drawn as part of the border. [*Example*: $\cancel{\boxed{a^2 + b^2 = c^2}}$ (Diagonal Strikethrough from Top Left) and $\overline{a^2 + b^2 = c^2}$ (no left or right edges). *end example*]

[*Example*:

$\underline{|x}$ (left and bottom edges) is represented by the following XML:

```
<m:borderBox>
  <m:borderBoxPr>
<m:hideTop m:val="1"/>
    <m:hideRight m:val="1"/>
  </m:borderBoxPr>
  <m:e>
    <m:r>
      <m:t>x</m:t>
```

```
      </m:r>
    </m:e>
  </m:borderBox>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_BorderBoxPr) is located in §A.6.1. *end note*]

22.1.2.13 box (Box Object)

This element specifies the box object, which is used to group components of an equation or other instance of mathematical text. A boxed object can (for example) serve as an operator emulator with or without an alignment point, serve as a line break point, have associated argSz, or be grouped such as not to allow line breaks within. If boxPr is omitted, all properties will be "false" by default.

[*Example*: The mathematical text $a == b$ uses a box around the double equal sign.

Its XML representation is as follows:

```
<m:r>
  <m:t>a</m:t>
</m:r>
<m:box>
  <m:boxPr>
    <m:opEmu    m:val="1"/>
    <m:aln/>
  </m:boxPr>
  <m:e>
      <m:r>
        <m:t>==</m:t>
      </m:r>
  </m:e>
</m:box>
<m:r>
  <m:t>b</m:t>
</m:r>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Box) is located in §A.6.1. *end note*]

22.1.2.14 boxPr (Box Properties)

This element specifies properties of the Box object, for example, whether the Box serves as operator emulator with or without an alignment point, serves as a line break point, or receives the correct spacing for the mathematical differential. [*Example*:

$a\ x\ b$ is partly represented by the following XML:

```
<m:box>
  <m:boxPr>
    <m:opEmu m:val="1"/>
  </m:boxPr>
  <m:e>
    <m:r>
      <m:t>x</m:t>
    </m:r>
  </m:e>
</m:box>
```

The x is set to be an operator emulator so it is given operator spacing around it.

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_BoxPr) is located in §A.6.1. *end note*]

22.1.2.15 brk (Break)

This element specifies whether there is a line break at the start of a run, or at the start of the Box object, such that the line wraps at the start of the run or box object. These user-defined line breaks occur when the XML tag `<m:brk/>` is encountered and does not follow a mathematical "order of precedence". If this element is omitted, a manual break is not inserted. In other words, when the element is absent, the default is for the parent structure to not manually break onto the next line. When the element is present and the val attribute is absent, the default of the val attribute is `0` meaning that this property's parent structure manually breaks onto the next line and is aligned with the beginning of the previous line.

The line may happen to wrap at this point if the mathematical text exceeds the column width. OMML does not specify behaviour for any kind of automatic line breaking. Instead it delegates this functionality to the underlying rendering engine similarly to how it does with the choice of operator spacing and the determination of whether an operator is unary or binary.

[*Example*: The following example includes a manual user-defined line break at the operator emulator:

$$a \\ == b$$

```
<m:r>
  <m:t>a</m:t>
</m:r>
```



```
<m:box>
  <m:boxPr>
    <m:opEmu    m:val="1"/>
    <m:brk/>
  </m:boxPr>
  <m:e>
    <m:r>
      <m:t>==</m:t>
    </m:r>
  </m:e>
</m:box>
<m:r>
  <m:t>b</m:t>
</m:r>
```

There is a break on the == symbol so a manual line break occurs right before the == symbol.

end example]

Attributes	Description
alnAt (Index of Operator to Align To)	Specifies the index of the operator on the previous line of mathematical text which shall be used as the alignment point for the current line of mathematical text . A line can be aligned to any operator on the previous line; this attribute specifies exactly which operator shall be the target of that alignment in cases where there are multiple operators. If alnAt is omitted, then all runs (r tag) that follow a brk tag will align with the left margin of the first run of mathematical text. [*Example*: For example, consider the break in this instance of mathematical text: $a + b + c + d + e$ $+ f + g$ The second line could theoretically be aligned to any of the four operators in the previous line. Specifying an alnAt value of 3 for the second line resolves this ambiguity; the second line is aligned to the third operator in the previous line. *end example*] The possible values for this attribute are defined by the ST_Integer255 simple type (§22.1.3.6).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ManualBreak) is located in §A.6.1. *end note*]

22.1.2.16 brkBin (Break on Binary Operators)

This element specifies how binary operators are treated when they coincide with a line break. If this element is omitted, the line break occurs before the binary operator. That is, the binary operator is the first element on the wrapped line.

Whether the element is absent or present without the val attribute, the default of the val attribute is `before`.

[*Example*: For example:

$$f(x) = a_{11} + a_{12} + \cdots + a_{nn}$$

before

$$f(x) = a_{11} + a_{12} + \cdots + a_{nn}$$

after

$$f(x) = a_{11} + a_{12} + \cdots + a_{nn}$$

repeat

The above examples demonstrate how such an equation would look depending on the value of the brkBin document-property. These actual examples do not rely on the brkBin property of this document to maintain their configurations. This would be impossible because the brkBin property can only be set to one value for an entire document at any given time. Here we are demonstrating all three.

The XML below demonstrates brkBin in use under mathPr:

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

Attributes	Description
val (Value)	Specifies where to break on binary operators. Possible values are `before`, `after`, and `repeat`. The possible values for this attribute are defined by the ST_BreakBin simple type (§22.1.3.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BreakBin) is located in §A.6.1. *end note*]

22.1.2.17 brkBinSub (Break on Binary Subtraction)

This element specifies how the subtraction operator is treated when it coincides with a line break, when brkBin is set to repeat. If this element is omitted, the subtraction operator is repeated before and after the break.

Whether the element is absent or present without the val attribute, the default of the val attribute is --.

[*Example*: The XML below demonstrates brkBinSub in use under mathPr:

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

[*Example*:

Given the following XML representation using brkBinSub:

```
<m:brkBinSub m:val="-+"/>
```

Because the document option for break on binary subtraction is set to -+, any breaks in Math paragraphs on subtraction operators will display a – on the line before the break and a + on the line after the break.

end example]

Attributes	Description
val (Value)	Specifies how the subtraction operator is treated when it coincides with a line break, when brkBin is set to repeat. Possible values are --, -+, and +-. The possible values for this attribute are defined by the ST_BreakBinSub simple type (§22.1.3.2).

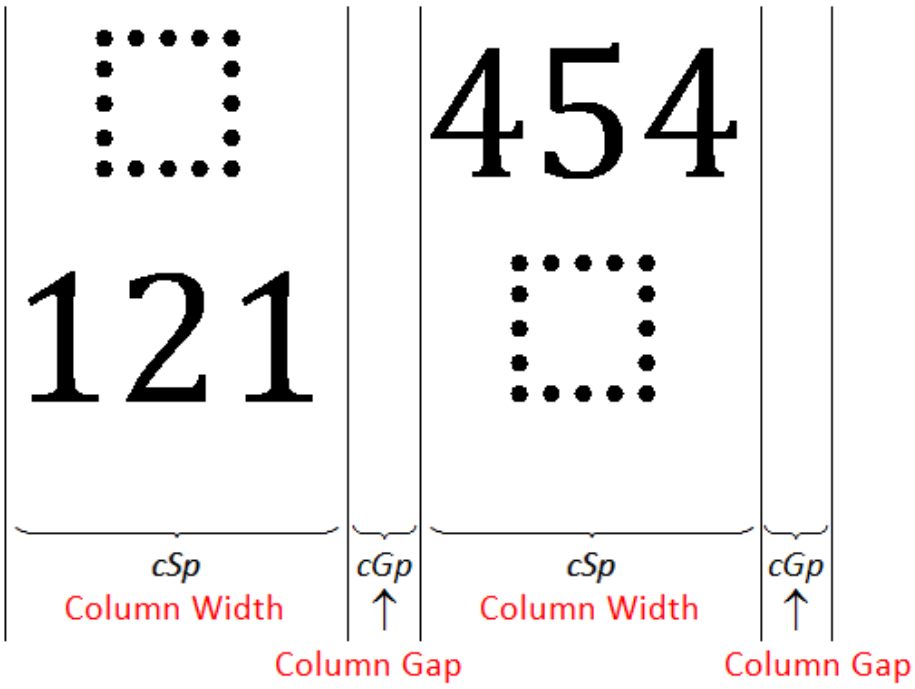
[*Note*: The W3C XML Schema definition of this element's content model (CT_BreakBinSub) is located in §A.6.1. *end note*]

22.1.2.18 cGp (Matrix Column Gap)

This element represents the (custom) column gap spacing information; the default value is 0 (which corresponds to 1 em). This value is interpreted differently depending on the value of cGpRule (§22.1.2.19). cGp is not used unless the value of cGpRule is 3 or 4. When cGpRule is omitted, the default spacing between matrix columns is 1 em (a val attribute value of 0).

Whether the element is absent or present without the val attribute, the default of the val attribute is 0.

The cGp gap spacing (also referred to as "Column Gap" or "Gap Width") is added to the cSp (Minimum Matrix Column Width) to determine the total Matrix Column Spacing (distance between the same edges of different columns). The following image depicts how cGp and cSp work together to define matrix column spacing in a 2x2 matrix:



[*Example*: With a cGpRule value of 4, the matrix: $\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}$ has 3 ems of spacing between columns (3 ems = 6 * 0.5 ems). The matrix properties that demonstrate this element in use are:

```
<m:mPr>
  <m:cGpRule  m:val="4"/>
  <m:cGp  m:val="6"/>
  <m:mcs>
    <m:mc>
      <m:mcPr>
        <m:count m:val="2"/>
        <m:mcJc m:val="center"/>

      </m:mcPr>
    </m:mc>
  </m:mcs>
</m:mPr>
```

end example]

See the documentation for cGpRule for more information about how cGp is interpreted for each value of cGpRule.

Attributes	Description
val (Value)	Specifies the amount of space between columns of the parent element (for cGp/cSp) or rows (for rSp). The manner in which this value is determined depends on the setting of the rule of the parent element. For cGp, if the rule is set to 3 (or "Exactly"), then the unit is interpreted as twips. If the rule is set to 4 (or "Multiple"), then the unit is interpreted as number of 0.5 em increments. For cSp, this value is measured in twips. There is no corresponding cSpRule. For rSp, if the rule is set to 3 (or "Exactly"), then the unit is interpreted as points. If the rule is set to 4 (or "Multiple"), then the unit is interpreted as half-lines. The possible values for this attribute are defined by the ST_UnSignedInteger simple type (§22.1.3.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnSignedInteger) is located in §A.6.1. *end note*]

22.1.2.19 cGpRule (Matrix Column Gap Rule)

This element specifies the type of gap (horizontal spacing) between columns of a matrix; the default is 0. Horizontal spacing units can be ems or points (stored as twips).

Whether the element is absent or present without the val attribute, the default of the val attribute is 0 (or single spacing gap).

[*Example*: The following matrix has double spacing (2 ems) between columns:

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}$$

The XML that represents this property in use is:

```
<m:mPr>
  <m:cGpRule  m:val="2"/>
  <m:mcs>
    <m:mc>
      <m:mcPr>
        <m:count  m:val="2"/>
        <m:mcJc  m:val="center"/>

      </m:mcPr>
    </m:mc>
  </m:mcs>
</m:mPr>
```

end example]

Attributes	Description
val (Value)	Specifies the type of spacing between rows and/or columns. Possible values are 0, 1, 2, 3, or 4, whose definitions are contained in the following table:

Value	Column/Row Gap	Example
0	Single spacing gap (1 em)	1 2
1	1.5 spacing gap (1.5 ems)	1 2
2	2 spacing gap (2 ems)	1 2
3	Exactly (for columns, rely on value of cGp, measured in twips) (for rows, rely on value of rSp, measured in points)	1 2
4	Multiple (for columns, rely on value of cGp, measured in 0.5 em increments) (for rows, rely on value of rSp, measured in lines)	1 2

Attributes	Description
	The possible values for this attribute are defined by the ST_SpacingRule simple type (§22.1.3.11).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SpacingRule) is located in §A.6.1. *end note*]

22.1.2.20 chr (Character)

This element specifies the character to be attached to the base of an accent object, a group character object, or an n-ary operator object. When the parent element is accPr, the chr value should be within the range of (U+0300–U+036F) or (U+20D0–U+20EF). When the parent element is groupChrPr, the chr value should be a horizontal stretch character, such as U+2190 (LEFTWARD ARROW). When the parent element is naryPr, the chr value should be an n-ary operator such as U+222B (INTEGRAL).

When the element is present and the val attribute is absent, the default of the val attribute is empty which means the corresponding objects character is absent.

If this property is omitted for accPr, the default accent character is U+0302 (ACCENT, COMBINING CIRCUMFLEX). If this property is omitted for groupChrPr, the default character is U+23DF (BOTTOM CURLY BRACKET). If this property is omitted for naryPr, the default character is U+222B (INTEGRAL).

In other words, when the element is absent, the defaults are as follows:

- accent object – Unicode character U+0302 (ACCENT, COMBINING CIRCUMFLEX)
- group-character object – Unicode character U+23DF (BOTTOM CURLY BRACKET)
- n-ary operator object – Unicode character U+222B (INTEGRAL)

[*Example*: Examples of accent characters are the dot, hat, and arrow in the following cases: $\dot{a}$ $\hat{a}$ $\vec{a}$.

For example, the following XML represents the acc $\tilde{a}$.

```
<m:acc>
  <m:accPr>
    <m:chr m:val="&#771;"/>
  </m:accPr>
  <m:e>
    <m:r>
      <m:t>a</m:t>
    </m:r>
  </m:e>
</m:acc>
```

end example]

Attributes	Description

val (value)	Specifies the character used by the parent element. When it is omitted, the parent uses its assigned default.
	
	[*Example*: Accent object ã:
	
	`<m:accPr>`
	`  <m:chr m:val="̃"/>`
	`</m:accPr>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Char simple type (§22.1.3.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Char) is located in §A.6.1. *end note*]

22.1.2.21 count (Matrix Column Count)

This element specifies the number of columns to which a property applies.

Whether the element is absent or present without the val attribute, the default of the val attribute is 1.

[*Example*: The example below specifies that two of the columns in the matrix described by the XML have the center property.

```
<m:mPr>
  <m:cSp  m:val="120"/>
  <m:mcs>
    <m:mc>
      <m:mcPr>
        <m:count  m:val="2"/>
        <m:mcJc m:val="center"/>

      </m:mcPr>
    </m:mc>
  </m:mcs>
</m:mPr>
```

end example]

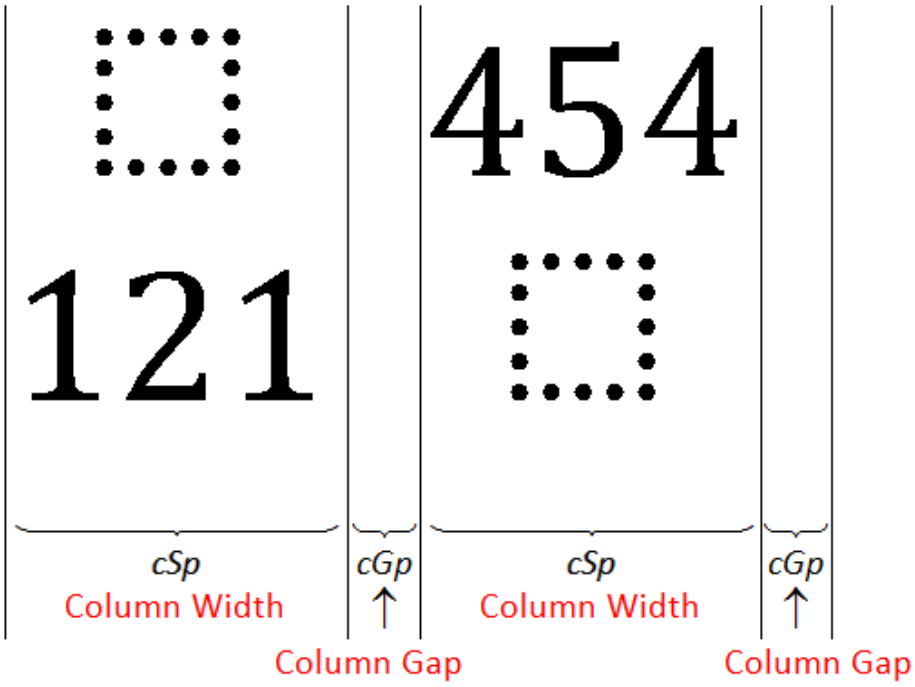
Attributes	Description
val (Value)	Specifies the number of columns to which a column property applies. [*Example*: A count attribute value of 3 specifies that the property applies to the next three columns of the matrix. *end example*]

Attributes	Description
	The possible values for this attribute are defined by the ST_Integer255 simple type (§22.1.3.6).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Integer255) is located in §A.6.1. *end note*]

22.1.2.22 cSp (Minimum Matrix Column Width)

This element specifies the minimum column width of a matrix. The actual column width of a matrix will be the greater of either the width of the column's widest argument or cSp. This additional spacing can be added to enhance appearance. If this element is omitted, the default minimum column width is 0. Whether the element is absent or present without the val attribute, the default of the val attribute is 0. The cGp gap spacing (also referred to as "Column Gap" or "Gap Width") is added to the cSp (Minimum Matrix Column Width) to determine the total Matrix Column Spacing (distance between the same edges of different columns). The value of cSp is interpreted as twips (a twip is 1/20th of a point). Therefore, a spacing of 1 point will be set by a cSp value of 20. This is the only use for cSp. There is no corresponding cSpRule. The following image depicts how cGp and cSp work together to define matrix column spacing in a 2x2 matrix:



[*Example*: The following XML specifies that there should never be fewer than 6 pts (120 twips) between adjacent column edges of the matrix:

1 2
3 4

```
<m:mPr>
  <m:cSp  m:val="120"/>
  <m:mcs>
    <m:mc>
      <m:mcPr>
        <m:count  m:val="2"/>
        <m:mcJc m:val="center"/>

      </m:mcPr>
    </m:mc>
  </m:mcs>
</m:mPr>
```

end example]

Attributes	Description
val (Value)	Specifies the amount of space between columns of the parent element (for cGp/cSp) or rows (for rSp). The manner in which this value is determined depends on the setting of the rule of the parent element. For cGp, if the rule is set to 3 (or "Exactly"), then the unit is interpreted as twips. If the rule is set to 4 (or "Multiple"), then the unit is interpreted as number of 0.5 em increments. For cSp, this value is measured in twips. There is no corresponding cSpRule. For rSp, if the rule is set to 3 (or "Exactly"), then the unit is interpreted as points. If the rule is set to 4 (or "Multiple"), then the unit is interpreted as half-lines. The possible values for this attribute are defined by the ST_UnSignedInteger simple type (§22.1.3.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnSignedInteger) is located in §A.6.1. *end note*]

22.1.2.23 ctrlPr (Control Properties)

This element specifies properties on control characters; that is, object characters that cannot be selected. Examples of control characters are n-ary operators (excluding their limits and bases), fraction bars (excluding the numerator and denominator), and grouping characters (excluding the base). ctrlPr allows formatting properties to be stored on these control characters. The control character inherits its formatting from the paragraph

formatting; ctrlPr contains the formatting differences between the control character and the paragraph formatting.

If this element is omitted, the character properties of the first control character are the same as the first character in the math object.

[*Example*: The example below shows that the control character is of font Cambria Math. All other formatting, such as text size and color, are the same as the paragraph.

```
<m:ctrlPr>
  <w:rPr>
    <w:rFonts w:ascii="Cambria Math" w:hAnsi="Cambria Math"/>
  </w:rPr>
</m:ctrlPr>
```

end example]

CtrlPr is also used to save properties on characters used in the built-down form of an instance of mathematical text that are not displayed in Professional form. For example, the mathematical text (in linear form) ∫_0^1 might have color on the _ or ^. Though these characters are not displayed in Professional form, their formatting is stored such that their formatting will roundtrip through build up and build down.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CtrlPr) is located in §A.6.1. *end note*]

22.1.2.24 d (Delimiter Object)

This element specifies the delimiter object, consisting of opening and closing delimiters (such as parentheses, braces, brackets, and vertical bars), and an element contained inside. The delimiter may have more than one element, with a designated separator character between each element. [*Example*:

Delimiter with one base: (x^2)

Delimiter with more than one base and separators, whose XML is shown below: $(x^2|y^2)$

```
<m:d>
  <m:e>
    <m:sSup>
      <m:e>
        <m:r>
          <m:t>x</m:t>
        </m:r>
      </m:e>
```

```
        <m:sup>
          <m:r>
            <m:rPr>
              <m:scr  m:val="roman"/>
              <m:sty  m:val="p"/>
            </m:rPr>
            <m:t>2</m:t>
          </m:r>
        </m:sup>
      </m:sSup>
    </m:e>
    <m:e>
      <m:sSup>
        <m:e>
          <m:r>
            <m:t>y</m:t>
          </m:r>
        </m:e>
        <m:sup>
          <m:r>
            <m:rPr>
              <m:scr m:val="roman"/>
              <m:sty  m:val="p"/>
            </m:rPr>
            <m:t>2</m:t>
          </m:r>
        </m:sup>
      </m:sSup>
    </m:e>
  </m:d>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_D) is located in §A.6.1. *end note*]

22.1.2.25 defJc (Default Justification)

This element specifies the default justification of display math, at the document level. Individual instances of mathematical text can overrule the default setting. If this element is omitted, the default justification is `centerGroup`. Whether the element is absent or present without the val attribute, the default of the val attribute is `centerGroup`.

Display math can be left justified, right justified, centered, or centered as a group. When display math is centered as a group, the mathematical text is left aligned within a block, and the entire block is centered with respect to column margins. If this element is omitted, the mathematical text is centered as a group.

[*Example*: The XML below demonstrates defJc in use:

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

Attributes	Description
val (Value)	Specifies the default justification of mathematical text in the document. Possible values are `center`, `centerGroup`, `left`, and `right`. The possible values for this attribute are defined by the ST_Jc simple type (§22.1.3.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathJc) is located in §A.6.1. *end note*]

22.1.2.26 deg (Degree)

This element specifies the degree in the mathematical radical. This element is optional. When omitted, the square root function, as in $\sqrt{x}$, is assumed. [*Example*:

The 3 in $\sqrt[3]{x}$ is represented by the XML below:

```
<m:rad>
  <m:deg>
     <m:r>
        <m:rPr>
          <m:scr   m:val="roman"/>
          <m:sty   m:val="p"/>
        </m:rPr>
        <m:t>3</m:t>
      </m:r>
  </m:deg>
```

```
    <m:e>
      <m:r>
        <m:t>x</m:t>
      </m:r>
    </m:e>
  </m:rad>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathArg) is located in §A.6.1. *end note*]

22.1.2.27 degHide (Hide Degree)

This element specifies the per-object option to hide the degree of a radical. Every rad has a deg, but the deg can appear or not appear. When degHide is set to 1 or true, the degree is not shown, as in $\sqrt{x}$ (XML shown below). When degHide is omitted, the default is 0 or false; that is, the degree is not hidden. In other words, when the element is absent, the default value of the property is 0 meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied.

[*Example*:

```
  <m:rad>
    <m:radPr>
      <m:degHide  m:val="1"/>
    </m:radPr>
    <m:deg>
    </m:deg>
    <m:e>
      <m:r>
        <m:t>x</m:t>
      </m:r>
    </m:e>
  </m:rad>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of 0 or false specifies that the property shall be explicitly turned off. This is

Attributes	Description
	implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.28 den (Denominator)

This element specifies the denominator of a fraction. [*Example*: For example, the b in $^{a}/_{b}$:

```
<m:f>
  <m:fPr>
    <m:type  m:val="skw"/>
  </m:fPr>
  <m:num>
    <m:r>
      <m:t>a</m:t>
    </m:r>
  </m:num>
  <m:den>
    <m:r>
      <m:t>b</m:t>
    </m:r>
  </m:den>
</m:f>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathArg) is located in §A.6.1. *end note*]

22.1.2.29 diff (Differential)

The element specifies the differential property on box. When `1` or `true`, the box acts as a differential (e.g., dx in an integrand), and receives the appropriate horizontal spacing for the mathematical differential. When this property is omitted, the box is not treated as a differential.

When the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied.

[*Example*: The following example demonstrates a box set as differential in use, both in its proper form and in XML:

$$\int_0^1 x\,dx$$

```
<m:nary>
  <m:naryPr>
    <m:chr m:val="&#8747;"/>
  </m:naryPr>
  <m:sub>
      <m:r>
        <m:rPr>
          <m:scr m:val="roman"/>
          <m:sty m:val="p"/>
        </m:rPr>
        <m:t>0</m:t>
      </m:r>
  </m:sub>
  <m:sup>
      <m:r>
        <m:rPr>
          <m:scr  m:val="roman"/>
          <m:sty  m:val="p"/>
        </m:rPr>
        <m:t>1</m:t>
      </m:r>
  </m:sup>
  <m:e>
    <m:r>
      <m:t>x</m:t>
    </m:r>
    <m:box>
      <m:boxPr>
        <m:diff  m:val="1"/>
      </m:boxPr>
      <m:e>
        <m:r>
          <m:t>dx</m:t>
        </m:r>
      </m:e>
    </m:box>
  </m:e>
</m:nary>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.30 dispDef (Use Display Math Defaults)

This element specifies the document-level property to overwrite paragraph settings for mathematical text. When omitted, this element is set to `1` or `true` and special math settings are applied. Whether the element is absent or present without the val attribute, the default of the val attribute is `1` meaning that this option is applied.

[*Example*: The XML below demonstrates dispDef in use:

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present.

Attributes	Description
	A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.31 dPr (Delimiter Properties)

This element specifies the properties of d, including the enclosing and separating characters, and the properties that affect the shape of the delimiters. [*Example*:

$[a + b]$ is represented by the following XML:

```
<m:d>
  <m:dPr>
    <m:begChr m:val="["/>
    <m:endChr m:val="]"/>
  </m:dPr>
  <m:e>
    <m:r>
      <m:t>a+b</m:t>
    </m:r>
  </m:e>
</m:d>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DPr) is located in §A.6.1. *end note*]

22.1.2.32 e (Element (Argument))

This tag, which is an abbreviation for "element", serves several functions (18 total) including that of the base argument of a mathematical object or function, the elements in an array, and the elements in boxes. If all subelements are omitted, this element specifies the presence of an empty argument. [*Example*: The numerator of this function is an empty argument: $\frac{}{2}$. *end example*] [*Note*: The tag was shortened to e to improve readability over "element". "Element" was preferred over "Base" because "Element" sounded more neutral and appropriate to its multiple uses. *end note*]

The following table shows the 18 different uses for e:

Parent Element	Use
acc	Accent base argument

Parent Element	Use
bar	Argument to which the bar is applied
borderBox	Argument around which the border box is drawn
box	Argument inside the abstract box
d	Argument inside the delimiters
eqArr	Each instance of mathematical text in the single-column array
func	Math argument list of the function
groupChr	Group character base
limLow	Base of the lower limit
limUpp	Base of the upper limit
mr	Each element in the matrix row
nary	n-ary and, e.g., integrand for an integral, summand for a summation
phant	Argument for the phantom
rad	Radicand
sPre	Base of the prescript object
sSub	Base of the subscript object
sSubSup	Base of the subsup object
sSup	Base of the superscript object

[*Example*: For example, the func $\sin x$ has fName sin and e x:

```
<m:func>
  <m:fName>
    <m:r>
      <m:t>sin</m:t>
    </m:r>
  </m:fName>
  <m:e>
    <m:r>
      <m:t>x</m:t>
    </m:r>
  </m:e>
</m:func>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathArg) is located in §A.6.1. *end note*]

22.1.2.33 endChr (Delimiter Ending Character)

This element specifies the ending, or closing, delimiter character. Mathematical delimiters are enclosing characters such as parentheses, brackets, and braces. If this element is omitted, the default endChr is ')'. In other words, when the element is absent, the default is for the delimiter object beginning character to be Unicode character U+0029 (RIGHT PARENTHESIS).

When the element is present and the val attribute is absent, the default of the val attribute is empty which means the delimiter object beginning character is absent.

[*Example*: In the following example, $\{a\}$ uses { and } as its enclosing characters:

```
<m:dPr>
  <m:begChr  m:val="{"/>
  <m:endChr m:val="}"/>
</m:dPr>
```

end example]

Attributes	Description

val (value)	Specifies the character used by the parent element. When it is omitted, the parent uses its assigned default.
	
	[*Example*: Accent object ã:
	
	`<m:accPr>`
	`  <m:chr m:val="̃"/>`
	`</m:accPr>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Char simple type (§22.1.3.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Char) is located in §A.6.1. *end note*]

22.1.2.34 eqArr (Array Object)

This element specifies the Array object (sometimes referred to as "Equation Array", despite its ability to hold mathematical text other than equations), an object consisting of one or more equations, expressions, or other mathematical text runs that can be vertically justified as a unit with respect to surrounding text on the line. Alignment of multiple points within each run of mathematical text can occur within the array through the use of align values and spacer values. An *align value* is an ampersand within the array which acts as an alignment point (as described in §22.1.2.3). A *spacer value* is an ampersand (represented by "&" in the example below) within the array which designates where space can be added in order to align the align values on different rows of the array. Within each argument in the array, every odd ampersand is an align value and every even

ampersand is a spacer value (as well, the beginning of each argument provides an implied spacer value). If eqArrPr is omitted then the default values for its properties will be used. These defaults are:

Property	Default Value
baseJc	“center”
ctrlPr	<The character property of the first control character will be the character property of the first character in the eqArr object >
maxDist	“0”
objDist	“0”
rSp	“0”
rSpRule	“0” <single>

[*Example*:

A simple array, $\begin{matrix} a = b + c \\ d + e = f \end{matrix}$, is represented as follows:

```
<m:eqArr>
  <m:e>
    <m:r>
      <m:t>a=b+c</m:t>
    </m:r>
  </m:e>
  <m:e>
    <m:r>
      <m:t>d+e=f</m:t>
    </m:r>
  </m:e>
</m:eqArr>
```

A more complex array with alignment points, e.g.:

$$\begin{aligned} x - \ y + \ z &= \ 10 \\ 3x + \ y + 2z &= \ 34 \\ -5x + 2y - \ z &= -14 \end{aligned}$$

is represented as follows:

```
<m:eqArr>
    <m:e>
        <m:r>
            <m:t>&amp;x+&amp;&amp;y+&amp;&amp;z&amp;&amp;= &amp;&amp;10</m:t>
        </m:r>
    </m:e>
```



```
      <m:e>
          <m:r>
              <m:t>3&amp;amp;x+&amp;amp;&amp;amp;y+&amp;amp;2&amp;amp;z&amp;amp;&amp;amp;= &amp;amp;&amp;amp;34</m:t>
          </m:r>
      </m:e>
      <m:e>
          <m:r>
              <m:t>-5&amp;amp;x+&amp;amp;2&amp;amp;y-&amp;amp;&amp;amp;z&amp;amp;&amp;amp;= &amp;amp;&amp;amp;-
  14</m:t>
          </m:r>
      </m:e>
  </m:eqArr>
```

Notice that the variables, operators, and digits of the sums line up properly, as the align values line up between each argument by adding space at the position of each spacer value.

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_EqArr) is located in §A.6.1. *end note*]

22.1.2.35 eqArrPr (Array Properties)

This element specifies the properties of the array object, including the vertical justification of the object and layout inside the object. [*Example*:

$$\begin{aligned} 111a &= c \\ 111a + 111b &= 111c \\ b &= 111c \end{aligned}$$

```
  <m:eqArr>
    <m:eqArrPr>
      <m:baseJc m:val="bottom"/>
    </m:eqArrPr>
    <m:e>
      <m:r>
        <m:t>111&amp;amp;a&amp;amp;&amp;amp;&amp;amp;&amp;amp;&amp;amp;&amp;amp;=&amp;amp;&amp;amp;c</m:t>
      </m:r>
    </m:e>
    <m:e>
      <m:r>
        <m:t>111&amp;amp;a&amp;amp;&amp;amp;+&amp;amp;111&amp;amp;b&amp;amp;&amp;amp;=&amp;amp;111&amp;amp;c</m:t>
      </m:r>
    </m:e>
    <m:e>
      <m:r>
```



```
        <m:t>&amp;&amp;&amp;&amp;&amp;b&amp;&amp;=&amp;111&amp;c</m:t>
      </m:r>
    </m:e>
  </m:eqArr>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_EqArrPr) is located in §A.6.1. *end note*]

22.1.2.36 f (Fraction Object)

This element specifies the fraction object, consisting of a numerator and denominator separated by a fraction bar. The fraction bar can be horizontal or diagonal, depending on the fraction properties. The fraction object is also used to represent the stack function, which places one element above another, with no fraction bar. [*Example*: Examples of fractions are:

Bar Fraction: $\frac{a}{b}$

Skewed Fraction: a/b

Linear Fraction: a/b

No-Bar Fraction (Stack): $\binom{n}{k}$

The fraction a/b is represented as:

```
<m:f>
  <m:fPr>
    <m:type  m:val="skw"/>
  </m:fPr>
  <m:num>
    <m:r>
      <m:t>a</m:t>
    </m:r>
  </m:num>
  <m:den>
    <m:r>
      <m:t>b</m:t>
    </m:r>
  </m:den>
</m:f>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_F) is located in §A.6.1. *end note*]

22.1.2.37 fName (Function Name)

This element specifies the name of the function in the Function-Apply object func. For example, function names are sin and cos.

[*Example*: As an example, the func $\sin x$ has fName sin and e x:

```
<m:func>
  <m:fName>
    <m:r>
      <m:t>sin</m:t>
    </m:r>
  </m:fName>
  <m:e>
    <m:r>
      <m:t>x</m:t>
    </m:r>
  </m:e>
</m:func>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathArg) is located in §A.6.1. *end note*]

22.1.2.38 fPr (Fraction Properties)

This element specifies the properties of the fraction object f. Properties of the Fraction object include the type or style of the fraction. The fraction bar can be horizontal or diagonal, depending on the fraction properties. The fraction object is also used to represent the stack function, which places one element above another, with no fraction bar. [*Example*: Examples of fractions are:

Bar Fraction: $\frac{a}{b}$

Skewed Fraction: a/b

Linear Fraction: a/b

No-Bar Fraction (Stack): $\binom{n}{k}$

The linear fraction c/d is represented by the following XML:

```
<m:f>
  <m:fPr>
    <m:type m:val="lin"/>
  </m:fPr>
  <m:num>
    <m:r>
```



```
        <m:t>c</m:t>
      </m:r>
    </m:num>
    <m:den>
      <m:r>
        <m:t>d</m:t>
      </m:r>
    </m:den>
  </m:f>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_FPr) is located in §A.6.1. *end note*]

22.1.2.39 func (Function Apply Object)

This element specifies the Function-Apply object, which consists of a function name and an argument element (e) acted upon. It is often applied using a form of linear format. For example, in the linear format described in Unicode Technical Article #28, this object is applied by using the Function Application character (U+2061). [*Example*: Examples of Function-Apply objects include: $\sin x$, $\tan^{-1} x^2$, and $\max_{0 \le x \le 1} xe^{-x^2}$.

As an example, the func $\sin x$ has fName sin and e x:

```
  <m:func>
    <m:fName>
      <m:r>
        <m:t>sin</m:t>
      </m:r>
    </m:fName>
    <m:e>
      <m:r>
        <m:t>x</m:t>
      </m:r>
    </m:e>
  </m:func>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Func) is located in §A.6.1. *end note*]

22.1.2.40 funcPr (Function Properties)

This element specifies properties such as ctrlPr that can be stored on the function apply object func. [*Example*:

$tan\ x$ is represented by the following XML:

```
  <m:func>
    <m:funcPr/>
```

```
    <m:fName>
      <m:r>
        <m:rPr>
          <m:sty m:val="p"/>
        </m:rPr>
        <m:t>tan</m:t>
      </m:r>
    </m:fName>
    <m:e>
      <m:r>
        <m:t>x</m:t>
      </m:r>
    </m:e>
  </m:func>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_FuncPr) is located in §A.6.1. *end note*]

22.1.2.41 groupChr (Group-Character Object)

This element specifies the Group-Character object, consisting of a character drawn above or below text, often with the purpose of visually grouping items. [*Example*: The following example demonstrates the groupChr in use, both in its proper form and in XML:

$$\overbrace{x+x+\cdots}$$

```
  <m:groupChr>
    <m:groupChrPr>
      <m:chr  m:val="&#9182;"/>
      <m:pos  m:val="top"/>
    </m:groupChrPr>
    <m:e>
      <m:r>
        <m:t>x+x+&#22EF;</m:t>
      </m:r>
    </m:e>
  </m:groupChr>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupChr) is located in §A.6.1. *end note*]

22.1.2.42 groupChrPr (Group-Character Properties)

This element specifies the properties of the Group-Character object groupChr. These properties can be used to specify the character placed above or below the argument, and the position of the character. When omitted, character ⏟ (U+23DF, BOTTOM CURLY BRACKET) is used as the chr and its pos is set to bot.

[*Example*:

$\overbrace{x+y}$ is represented by the following XML:

```
<m:groupChr>
  <m:groupChrPr>
    <m:chr m:val="⏞"/>
    <m:pos m:val="top"/>
    <m:vertJc m:val="bot"/>
  </m:groupChrPr>
  <m:e>
    <m:r>
      <m:t>x+y</m:t>
    </m:r>
  </m:e>
</m:groupChr>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupChrPr) is located in §A.6.1. *end note*]

22.1.2.43 grow (n-ary Grow)

This element specifies the growth property of n-ary operators. When `0` or `false`, n-ary operators such as integrals and summations do not grow to match the size of their operand height. When `1` or `true`, the n-ary operator grows vertically to match its operand height. If this property is omitted, grow is set to `0`.

When the element is absent, the default value of the property when it is a child of a delimiter object is `1` (meaning that this property is applied) and when it is a child of an n-ary operator object it is `0` (meaning that this property is not applied).

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

[*Example*: The two integrals below demonstrate the difference between grow = `0` and grow = `1`.

$$\int_0^1 \frac{x^2}{x+y}dx \int_0^1 \frac{y^2}{x+y}dy$$

The XML that defines nary growth is:

```
<m:naryPr>
  <m:chr  m:val="&#8747;"/>
  <m:grow m:val="1"/>
</m:naryPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.44 hideBot (Hide Bottom Edge)

This element specifies the hidden or shown state of the bottom edge of borderBox. When this element is omitted, the bottom edge is shown. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

When applied, the bottom border is hidden, as in $\overline{|abc|}$. [*Example*:

```
<m:borderBox>
  <m:borderBoxPr>
    <m:hideBot/>
  </m:borderBoxPr>
  <m:e>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:borderBox>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.45 hideLeft (Hide Left Edge)

This element specifies the hidden or shown state of the left edge of borderBox. When this element is omitted, the edge is shown. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

When applied, the left border is hidden, as in $\overline{abc}$. [*Example*:

```
<m:borderBox>
  <m:borderBoxPr>
    <m:hideLeft/>
  </m:borderBoxPr>
  <m:e>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:borderBox>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is

Attributes	Description
	implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.46 hideRight (Hide Right Edge)

This element specifies the hidden or shown state of the right edge of borderBox. When this element is omitted, the edge is shown. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

When applied, the right border is hidden, as in $\boxed{abc}$.

[*Example*:

```
<m:borderBox>
  <m:borderBoxPr>
    <m:hideRight/>
  </m:borderBoxPr>
  <m:e>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:borderBox>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.47 hideTop (Hide Top Edge)

This element specifies the hidden or shown state of the top edge of borderBox. When this element is omitted, the edge is shown. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

When applied, the top border is hidden, as in $\boxed{abc}$. [*Example*:

```
<m:borderBox>
  <m:borderBoxPr>
    <m:hideTop/>
  </m:borderBoxPr>
  <m:e>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:borderBox>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.48 interSp (Inter-Equation Spacing)

This element specifies spacing between equations, expressions, or other instances of mathematical text within a display math paragraph, in twips.

Attributes	Description
val (Value)	Specifies the value, in twips, of the parent element. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.6.1. *end note*]

22.1.2.49 intLim (Integral Limit Locations)

This element specifies the document setting for the default placement of integral limits, when converted from a linear form to a two-dimensional output (professional form). Limits can be either centered above and below the integral, or positioned just to the right of the operator, as in:

$$\int_{a}^{b} x\,dx \quad \int_{a}^{b} x\,dx$$

When this integral object is written linearly, as ∫ _a^b, the placement of limits is ambiguous. intLim is a document-level property that specifies the default positioning. When this element is omitted, the default placement of integral limits is subSup (that is, the location of subscripts and superscripts, or just to the right of the base or operator). Whether the element is absent or present without the val attribute, the default of the val attribute is subSup.

[*Example*: The XML that specifies this property in use is:

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

Attributes	Description
val (Value)	Specifies the default location of limits on the parent object. Possible values are subSup

Attributes	Description
	and undOvr. The possible values for this attribute are defined by the ST_LimLoc simple type (§22.1.3.8).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LimLoc) is located in §A.6.1. *end note*]

22.1.2.50 intraSp (Intra-Equation Spacing)

This element specifies the spacing between adjacent display math paragraphs, in twips. If this element is omitted, no spacing is applied between adjacent math paragraphs.

Attributes	Description
val (Value)	Specifies the value, in twips, of the parent element. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.6.1. *end note*]

22.1.2.51 jc (Justification)

This element specifies justification of the math paragraph (a series of adjacent instances of mathematical text within the same paragraph). A math paragraph can be Left Justified, Right Justified, Centered, or Centered as Group. If this element is omitted, the math paragraph is Centered as Group. Whether the element is absent or present without the val attribute, the default of the val attribute is centerGroup. This means that the instances of mathematical text can be aligned with respect to each other, but the entire group of mathematical text is centered as a whole. [*Example*: An example of Centered as Group is the following example, in which each equation is left-aligned, but the series is centered:

$$x = x_1 + x_2 + x_3 + \cdots$$
$$y = y_1 + y_2 + y_3 + y_4 + \cdots$$
$$z = z_1 + z_2 + z_3 + z_4 + z_5 + \cdots$$

The XML that demonstrates jc in use is:

```
<m:oMathParaPr>
  <m:jc m:val="centerGroup"/>
</m:oMathParaPr>
```

end example]

Attributes	Description
val (Value)	Specifies the default justification of mathematical text in the document. Possible values are `center`, `centerGroup`, `left`, and `right`. The possible values for this attribute are defined by the ST_Jc simple type (§22.1.3.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathJc) is located in §A.6.1. *end note*]

22.1.2.52 lim (Limit)

This element specifies the lower limit of the limLow object and the upper limit of the limUpp function.
[*Example*: For example, the limit of the limLow $\lim_{n \to \infty}$ is $n \to \infty$. The XML that specifies this object is:

```
<m:limLow>
  <m:e>
    <m:r>
      <m:rPr>
        <m:scr m:val="roman"/>
        <m:sty  m:val="p"/>
      </m:rPr>
      <m:t>lim</m:t>
    </m:r>
  </m:e>
  <m:lim>
    <m:r>
      <m:t>n&#8594;&#8734;</m:t>
    </m:r>
  </m:lim>
</m:limLow>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathArg) is located in §A.6.1. *end note*]

22.1.2.53 limLoc (n-ary Limit Location)

This element specifies the location of limits in n-ary operators. Limits can be either centered above and below the n-ary operator (shown in the first summation below), or positioned just to the right of the operator (shown in the second summation below).

$$\sum_{i=0}^{n} x_n \sum_{i=0}^{n} x_n$$

When the element is absent, the default value of this property is ignored and the intLim (for integral n-ary operator objects) and naryLim (for all other n-ary operator objects) options are looked to for the limit placement.

Ultimately, when this element is omitted, the default location is undOvr in display mode and subSup otherwise. When the element is present and the val attribute is absent, the default of the val attribute undOvr. [*Example*: The XML representing the relevant properties of the second summation above (subSup) is:

```
<m:naryPr>
  <m:chr  m:val="&#8721;"/>
  <m:limLoc m:val="subSup"/>
  <m:grow m:val="1"/>
</m:naryPr>
```

end example]

Attributes	Description
val (Value)	Specifies the default location of limits on the parent object. Possible values are subSup and undOvr. The possible values for this attribute are defined by the ST_LimLoc simple type (§22.1.3.8).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LimLoc) is located in §A.6.1. *end note*]

22.1.2.54 limLow (Lower-Limit Object)

This element specifies the Lower-Limit object, consisting of text on the baseline and reduced-size text immediately below it. Examples of limLow include $\lim_{n\to\infty}$ and $\max_{0\le x\le 1}$. [*Example*: The XML that represents $\lim_{n\to\infty}$ is:

```
<m:limLow>
  <m:e>
    <m:r>
      <m:rPr>
        <m:scr m:val="roman"/>
        <m:sty  m:val="p"/>
      </m:rPr>
      <m:t>lim</m:t>
    </m:r>
  </m:e>
```

```
    <m:lim>
      <m:r>
        <m:t>n&#8594;&#8734;</m:t>
      </m:r>
    </m:lim>
  </m:limLow>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_LimLow) is located in §A.6.1. *end note*]

22.1.2.55 limLowPr (Lower-Limit Properties)

This element specifies control properties (ctrlPr) that can be stored on the Lower Limit (limLow).

[*Example*:

$\lim_{a\to\infty}$ is represented by the following XML:

```
  <m:limLow>
    <m:limLowPr/>
    <m:e>
      <m:r>
        <m:t>lim</m:t>
      </m:r>
    </m:e>
    <m:lim>
      <m:r>
        <m:t>a→∞</m:t>
      </m:r>
    </m:lim>
  </m:limLow>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_LimLowPr) is located in §A.6.1. *end note*]

22.1.2.56 limUpp (Upper-Limit Object)

This element specifies the Upper-Limit object, consisting of text on the baseline and reduced-size text immediately above it. [*Example*: Examples of limUpp include $\overbrace{x+x+x}^{k \text{ times}}$ and $\overset{\text{def}}{=}$.

The XML that specifies the $\text{limUpp}\overset{\text{def}}{=}$ is:


```
<m:limUpp>
  <m:e>
    <m:r>
      <m:t>=</m:t>
    </m:r>
  </m:e>
  <m:lim>
    <m:r>
      <m:rPr>
        <m:nor/>
      </m:rPr>
      <m:t>def</m:t>
    </m:r>
  </m:lim>
</m:limUpp>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_LimUpp) is located in §A.6.1. *end note*]

22.1.2.57 limUppPr (Upper-Limit Properties)

This element specifies control properties (ctrlPr) that can be stored on the Upper Limit (limUpp).

[*Example*:

$\lim_{a \to \infty}$ is represented by the following XML:

```
<m:limUpp>
  <m:limUppPr/>
  <m:e>
    <m:r>
      <m:t>lim</m:t>
    </m:r>
  </m:e>
  <m:lim>
    <m:r>
      <m:t>a→∞</m:t>
    </m:r>
  </m:lim>
</m:limUpp>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_LimUppPr) is located in §A.6.1. *end note*]

22.1.2.58 lit (Literal)

This element specifies that the characters in the run are literal; that is, they are to be interpreted literally and not be built up based on any implied mathematical meaning. This is especially useful for operators or other special characters that signal a need for build up to an OMML reader. These characters are often encountered during a given instance of mathematical text when presented in a 1-dimensional linear format, such as the linear format defined by Unicode Technical Note #28 (Sargent 2006).

When the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

[*Example*: In the following XML, the + operator is treated literally and does not receive proper binary spacing:

```
<m:r>a</m:r>
<m:r>
  <m:rPr>
    <m:lit/>
  </m:rPr>
  <m:t>+</m:t>
</m:r>
<m:r>
  <m:t>b</m:t>
</m:r>
```

As another example, consider the following Office Open XML Math markup:

```
<m:r>
    <m:t>a</m:t>
</m:r>
<m:r>
    <m:rPr>
        <m:lit/>
    </m:rPr>
    <m:t>/</m:t>
</m:r>
<m:r>
    <m:t>b</m:t>
</m:r>
```

This would be displayed as: a/b. If built up, it would normally be converted to $\frac{a}{b}$; however, the presence of this element specifies that the solidus should be treated literally, resulting in a/b.

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.59 lMargin (Left Margin)

This element specifies the left margin for math, in twips. If this element is omitted, no left margin is used. In other words, when the element is absent, the default value of the option is `0`.

When the element is present and the val attribute is absent, the default of the val attribute is `1440` (or 1 inch).

Math margins are added to the paragraph settings for margins. If the sum of lMargin and rMargin exceed the width available, lMargin should be ignored. [*Example*: The following XML demonstrates an lMargin setting of 1".

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="1440"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

Attributes	Description
val (Value)	Specifies the value, in twips, of the parent element.

Attributes	Description
	The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.6.1. *end note*]

22.1.2.60 m (Matrix Object)

This element specifies the Matrix object, consisting of one or more elements laid out in one or more rows and one or more columns. It is important to note that matrices do not have built in delimiters. Like other math elements, matrices are contained in a delimiter object (§22.1.2.24) when delimiters are desired. Empty arguments (see §22.1.2.32) can be used to create gaps in matrices. The plcHide tag (§22.1.2.83) can be used to indicate whether the empty arguments should be visible in the matrix (see the plcHide documentation for more information). If mPr is omitted, the values of baseJc, cGp, cGpRule, cSp, ctrlPr, mcs, plcHide, rSp, and rSpRule are shown in the following table:

Property	Default Value
baseJc	"center"
cGp	"0"
cGpRule	"0" <single>
cSp	"0"
ctrlPr	<The character property of the first control character shall be the character property of the first character in the m object >
mcs	<All columns will be vertically center aligned>
plcHide	"0"
rSp	"0"
rSpRule	"0" <single>

[*Example*: Examples of matrices are: $\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$ and $\begin{bmatrix} 1 & \\ & 1 \end{bmatrix}$. Below is an example of a 2x2 matrix and its XML in its proper form (surrounding delimiters are not a property of the matrix and are only shown to demonstrate their relationship with a matrix. See the documentation for the d tag for XML examples of delimiters).

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

```
<m:d>
```



```
<m:m>
  <m:mPr>
    <m:mcs>
      <m:mc>
        <m:mcPr>
          <m:count m:val="2"/>
          <m:mcJc  m:val="center"/>

        </m:mcPr>
      </m:mc>
    </m:mcs>
  </m:mPr>
  <m:mr>
    <m:e>
      <m:r>
        <m:rPr>
          <m:scr  m:val="roman"/>
          <m:sty  m:val="p"/>
        </m:rPr>
        <m:t>1</m:t>
      </m:r >
    </m:e>
    <m:e>
      <m:r>
        <m:rPr>
          <m:scr  m:val="roman"/>
          <m:sty  m:val="p"/>
        </m:rPr>
        <m:t>2</m:t>
      </m:r >
    </m:e>
  </m:mr>
  <m:mr>
    <m:e>
      <m:r>
        <m:rPr>
          <m:scr  m:val="roman"/>
          <m:sty  m:val="p"/>
        </m:rPr>
        <m:t>3</m:t>
      </m:r >
    </m:e>
```

```
        <m:e>
          <m:r>
            <m:rPr>
              <m:scr  m:val="roman"/>
              <m:sty  m:val="p"/>
            </m:rPr>
            <m:t>4</m:t>
          </m:r >
        </m:e>
      </m:mr>
    </m:m>
  </m:d>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_M) is located in §A.6.1. *end note*]

22.1.2.61 mathFont (Math Font)

This element specifies the default math font to be used in the document. If this element is omitted, font substitution (§17.8.2) should be used to determine the most appropriate font for use throughout the document. [*Example*: The XML containing this property is:

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="1440"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

[*Example*:

Given the following equation:

$a + b = c$

and the following XML:


```
<m:mathFont m:val="Cambria Math"/>
```

Because the document option for Math font is set to the mathematically enabled font Cambria Math, Math is formatted with that font.

end example]

Attributes	Description
val (value)	Specifies the default math font to be used in the document. The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.6.1. *end note*]

22.1.2.62 mathPr (Math Properties)

This element specifies the document-level properties for all math in the document.

[*Example*:

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

These are the default settings for the math properties element and its children.

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_MathPr) is located in §A.6.1. *end note*]

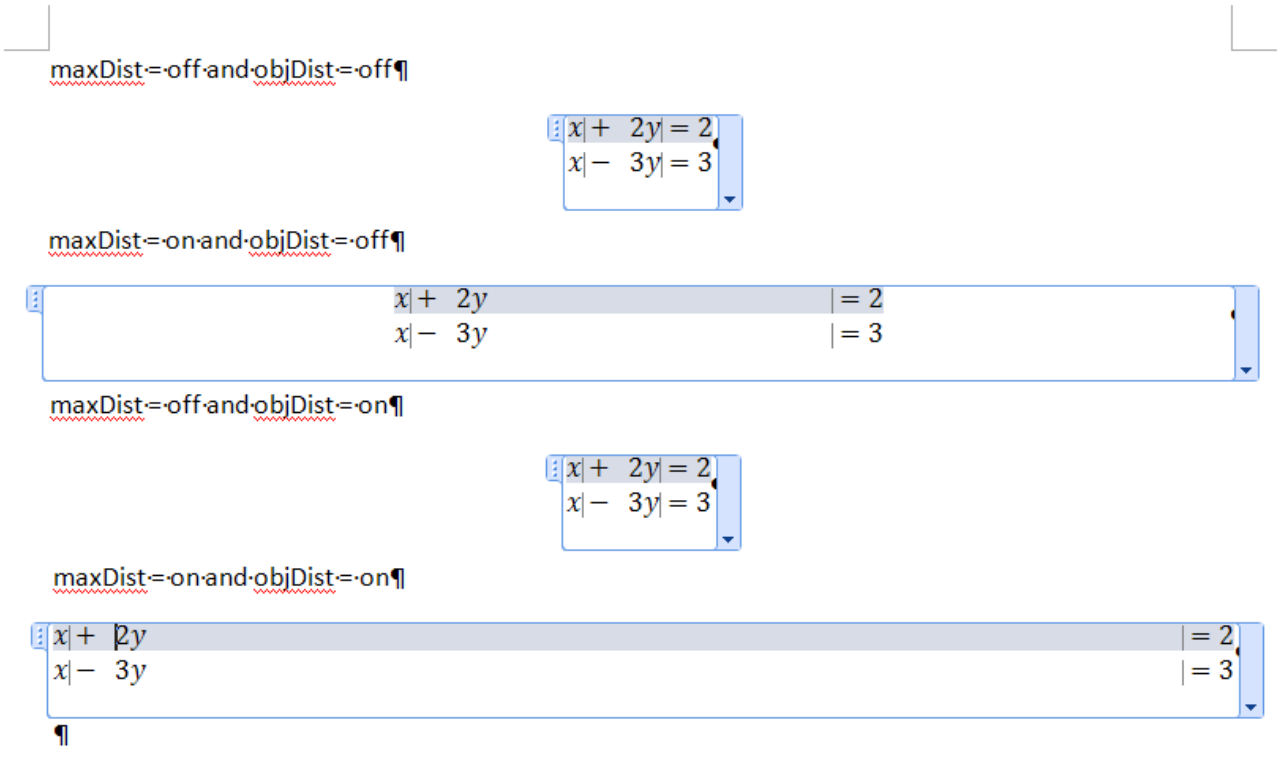
22.1.2.63 maxDist (Maximum Distribution)

This element specifies Array Maximum Distribution. When `1` or `true`, the array is spaced to the maximum width of the containing element (page, column, cell, etc.). The example image below illustrates an array expanded to fit the page, which is the containing element in this example. The maxDist option is commonly used with the objDist option. The objDist option is used to expand the distribution of mathematical text within the bounds of an array while not impacting the Array Distribution itself.

When the maxDist element is omitted, Array Maximum Distribution is off. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

[*Example*: The following diagram illustrates all four possible combinations of maxDist and objDist:



XML for the fourth example shown in the above image is:

```
<m:oMathPara>
  <m:oMath>
    <m:eqArr>
      <m:eqArrPr>
        <m:maxDist m:val="1" />
```

```
          <m:objDist m:val="1" />
        </m:eqArrPr>
        <m:e>
          <m:r>
            <m:t>x</m:t>
          </m:r>
          <m:r>
             <m:t>&+</m:t>
          </m:r>
          <m:r>
            <m:t>2y&=2</m:t>
          </m:r>
        </m:e>
        <m:e>
          <m:r>
            <m:t>x</m:t>
          </m:r>
          <m:r>
            <m:t>&-</m:t>
          </m:r>
          <m:r>
            <m:t>3y&=3</m:t>
          </m:r>
        </m:e>
      </m:eqArr>
    </m:oMath>
  </m:oMathPara>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.64 mc (Matrix Column)

This element specifies a single column in a matrix m. [*Example*: An example of this element in use is:

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

```
<m:m>
  <m:mPr>
    <m:mcs>
      <m:mc>
        <m:mcPr>
          <m:count m:val="2"/>
          <m:mcJc  m:val="center"/>

        </m:mcPr>
      </m:mc>
    </m:mcs>
  </m:mPr>
  <m:mr>
    <m:e>
      <m:r>
        <m:rPr>
          <m:scr  m:val="roman"/>
          <m:sty  m:val="p"/>
        </m:rPr>
        <m:t>1</m:t>
      </m:r >
    </m:e>
    <m:e>
      <m:r>
        <m:rPr>
          <m:scr  m:val="roman"/>
          <m:sty  m:val="p"/>
        </m:rPr>
        <m:t>2</m:t>
      </m:r >
    </m:e>
  </m:mr>
```



```
    <m:mr>
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>3</m:t>
        </m:r >
      </m:e>
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>4</m:t>
        </m:r >
      </m:e>
    </m:mr>
  </m:m>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_MC) is located in §A.6.1. *end note*]

22.1.2.65 mcJc (Matrix Column Justification)

This element specifies the justification of a matrix column (or group of matrix columns) mc. When this element is omitted, the column is centered. Whether the element is absent or present without the val attribute, the default of the val attribute is `center`. The matrix below has three columns. The leftmost column is left-justified, the rightmost column is right-justified, and the center column is centered: $\begin{pmatrix} 1 & 1 & 1 \\ 23 & 23 & 23 \\ 456 & 456 & 456 \end{pmatrix}$

[*Example*: A simple example of this property in use is a 2x2 matrix with both columns centered:

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

```
  <m:m>
    <m:mPr>
      <m:mcs>
        <m:mc>
          <m:mcPr>
```

```
            <m:count m:val="2"/>
            <m:mcJc  m:val="center"/>

          </m:mcPr>
        </m:mc>
      </m:mcs>
    </m:mPr>
    <m:mr>
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>1</m:t>
        </m:r >
      </m:e>
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>2</m:t>
        </m:r >
      </m:e>
    </m:mr>
    <m:mr>
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>3</m:t>
        </m:r >
      </m:e>
```



```
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>4</m:t>
        </m:r >
      </m:e>
    </m:mr>
  </m:m>
```

end example]

Attributes	Description

val (Value)	Specifies the horizontal alignment of the parent element. Possible values are `left`, `right`, and `center`. [*Example*:
	`<m:mcPr>`
	`  <m:mcJc  m:val="center"/>`
	`  <m:count m:val="2"/>`
	`</m:mcPr>`
	
	The possible values for this attribute are defined by the ST_XAlign simple type (§22.9.2.18).

[*Note*: The W3C XML Schema definition of this element's content model (CT_XAlign) is located in §A.6.1. *end note*]

22.1.2.66 mcPr (Matrix Column Properties)

This element specifies the properties of the matrix column mn, including the number of columns and the type of justification. [*Example*: As an extreme example, the following matrix has two columns that are left justified (count is 2) and three columns that are right justified (count is 3). $\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 23 & 23 & 23 & 23 & 23 \\ 456 & 456 & 456 & 456 & 456 \end{pmatrix}$ *end example*]

[*Example*:

$\begin{matrix} 1 & 2222 & 3 \\ 4444 & 5 & 6666 \end{matrix}$ is represented by the following XML:

```
  <m:m>
    <m:mPr>
      <m:mcs>
        <m:mc>
```

```
          <m:mcPr>
            <m:count m:val="3"/>
            <m:mcJc m:val="right"/>
          </m:mcPr>
        </m:mc>
      </m:mcs>
    </m:mPr>
    <m:mr>
      <m:e>
        <m:r>
          <m:t>1</m:t>
        </m:r>
      </m:e>
      <m:e>
        <m:r>
          <m:t>2222</m:t>
        </m:r>
      </m:e>
      <m:e>
        <m:r>
          <m:t>3</m:t>
        </m:r>
      </m:e>
    </m:mr>
    <m:mr>
      <m:e>
        <m:r>
          <m:t>4444</m:t>
        </m:r>
      </m:e>
      <m:e>
        <m:r>
          <m:t>5</m:t>
        </m:r>
      </m:e>
      <m:e>
        <m:r>
          <m:t>6666</m:t>
        </m:r>
      </m:e>
    </m:mr>
  </m:m>
```

All three columns are right aligned.

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_MCPr) is located in §A.6.1. *end note*]

22.1.2.67 mcs (Matrix Columns)

This element specifies the collection of columns of the matrix m. [*Example*: An example of this element in use is:

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

```
<m:m>
  <m:mPr>
    <m:mcs>
      <m:mc>
        <m:mcPr>
          <m:count m:val="2"/>
          <m:mcJc  m:val="center"/>

        </m:mcPr>
      </m:mc>
    </m:mcs>
  </m:mPr>
  <m:mr>
    <m:e>
      <m:r>
        <m:rPr>
          <m:scr  m:val="roman"/>
          <m:sty  m:val="p"/>
        </m:rPr>
        <m:t>1</m:t>
      </m:r >
    </m:e>
    <m:e>
      <m:r>
        <m:rPr>
          <m:scr  m:val="roman"/>
          <m:sty  m:val="p"/>
        </m:rPr>
        <m:t>2</m:t>
      </m:r >
    </m:e>
  </m:mr>
```

```
    <m:mr>
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>3</m:t>
        </m:r >
      </m:e>
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>4</m:t>
        </m:r >
      </m:e>
    </m:mr>
  </m:m>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_MCS) is located in §A.6.1. *end note*]

22.1.2.68 mPr (Matrix Properties)

This element specifies properties of the matrix m, including the justification of the matrix and the layout of elements within the matrix. [*Example*:

$a+1 \quad 2+b$ is represented by the following XML:
$3 \quad 4$

```
  <m:oMath>
    <m:r>
      <m:t>a+</m:t>
    </m:r>
    <m:m>
      <m:mPr>
        <m:baseJc m:val="top"/>
      </m:mPr>
      <m:mr>
        <m:e>
          <m:r>
            <m:t>1</m:t>
```



```
          </m:r>
        </m:e>
        <m:e>
          <m:r>
            <m:t>2</m:t>
          </m:r>
        </m:e>
      </m:mr>
      <m:mr>
        <m:e>
          <m:r>
            <m:t>3</m:t>
          </m:r>
        </m:e>
        <m:e>
          <m:r>
            <m:t>4</m:t>
          </m:r>
        </m:e>
      </m:mr>
    </m:m>
    <m:r>
      <m:t>+b</m:t>
    </m:r>
  </m:oMath>
```

This top row of the matrix is aligned with the rest of the baseline.

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_MPr) is located in §A.6.1. *end note*]

22.1.2.69 mr (Matrix Row)

This element specifies a single row of the matrix m. [*Example*: An example of this element in use is the following example, a 2x2 matrix. There are two rows; the first contains the elements 1 and 2; the second contains 3 and 4.

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

```
<m:m>
  <m:mPr>
    <m:mcs>
      <m:mc>
        <m:mcPr>
```

```
            <m:count m:val="2"/>
            <m:mcJc  m:val="center"/>

          </m:mcPr>
        </m:mc>
      </m:mcs>
    </m:mPr>
    <m:mr>
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>1</m:t>
        </m:r >
      </m:e>
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>2</m:t>
        </m:r >
      </m:e>
    </m:mr>
    <m:mr>
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>3</m:t>
        </m:r >
      </m:e>
      <m:e>
```



```
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>4</m:t>
        </m:r >
      </m:e>
    </m:mr>
  </m:m>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_MR) is located in §A.6.1. *end note*]

22.1.2.70 nary (n-ary Operator Object)

This element specifies an n-ary object, consisting of an n-ary object, a base (or operand), and optional upper and lower limits. Examples of n-ary objects are: $\int_0^1 x\,dx$, $\sum_k \binom{n}{k}$, $\prod_{k=1}^{n} A_k$, and $\cup_{=1}^{m}(X_n \cap Y_n)$. [*Example*: The example below demonstrates an n-ary object in its proper form and XML representation:

$$\int_0^1 x\,dx$$

```
<m:nary>
  <m:naryPr>
    <m:chr m:val="&#8747;"/>
  </m:naryPr>
  <m:sub>
      <m:r>
        <m:rPr>
          <m:scr m:val="roman"/>
          <m:sty m:val="p"/>
        </m:rPr>
        <m:t>0</m:t>
      </m:r>
  </m:sub>
```

```
    <m:sup>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>1</m:t>
        </m:r>
    </m:sup>
    <m:e>
      <m:r>
        <m:t>x</m:t>
      </m:r>
      <m:box>
        <m:boxPr>
          <m:diff  m:val="1"/>
        </m:boxPr>
        <m:e>
          <m:r>
            <m:t>dx</m:t>
          </m:r>
        </m:e>
      </m:box>
    </m:e>
  </m:nary>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Nary) is located in §A.6.1. *end note*]

22.1.2.71 naryLim (n-ary Limit Location)

This element specifies the document setting for the default placement of n-ary limits other than integrals (since integrals are most often written as subSup and other n-ary operators are most often written as undOvr), when converted from a built down form to a two-dimensional output (professional form). Limits can be either centered above and below the n-ary operator, or positioned just to the right of the operator, as in:

$$\sum_{i=0}^{n} x_i \quad \sum_{i=0}^{n} x_i$$

When this summation object is written in built down form, as in $\sum_(i = 0)\hat{}n$, for example, the placement of limits is ambiguous. naryLim specifies this positioning. When this element is omitted, the default placement of n-ary limits is undOvr (that is, the location of lower and upper limits). Whether the element is absent or present without the val attribute, the default of the val attribute is undOvr.

[*Example*: An example XML of this element in use is:


```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

Attributes	Description
val (Value)	Specifies the default location of limits on the parent object. Possible values are subSup and undOvr. The possible values for this attribute are defined by the ST_LimLoc simple type (§22.1.3.8).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LimLoc) is located in §A.6.1. *end note*]

22.1.2.72 naryPr (n-ary Properties)

This element specifies the properties of the n-ary object.

These include the n-ary operator character that is used (using the chr element), the shape and height of the operator (using the grow element), the location of limits (using the limLoc element), and whether empty limits display a placeholder character or are hidden (using the subHide and supHide elements). [*Example*: N-ary operators include: $\sum$, $\int$, $\cup$, $\wedge$. *end example*]

[*Example*:

$\sum_{1}^{20} x$ is represented by the following XML:

```
<m:nary>
  <m:naryPr>
    <m:chr m:val="∑"/>
    <m:limLoc m:val="subSup"/>
  </m:naryPr>
  <m:sub>
```

```
      <m:r>
        <m:t>1</m:t>
      </m:r>
    </m:sub>
    <m:sup>
      <m:r>
        <m:t>20</m:t>
      </m:r>
    </m:sup>
    <m:e>
      <m:r>
        <m:t>x</m:t>
      </m:r>
    </m:e>
  </m:nary>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NaryPr) is located in §A.6.1. *end note*]

22.1.2.73 noBreak (No Break)

This property specifies the "unbreakable" property on the Box object box. When `1` or `true`, no line breaks can occur within the box. This can be important for operator emulators that consist of more than one binary operator. When this element is not specified, breaks can occur inside box. Whether the element is absent or present without the val attribute, the default of the val attribute is `1` meaning that this property is applied.

[*Example*: Sample XML containing this element is below. In this box, breaks are allowed.

```
<m:boxPr>
  <m:noBreak m:val="0"/>
</m:boxPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.74 nor (Normal Text)

This element specifies that the run is normal text, i.e., math italics and math spacing are not applied. In a normal text run, no characters will trigger reformatting of a linear expression into a two-dimensional expression.

When the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

[*Example*: The example below illustrates three runs of normal text, along with the XML representation of the formula:

$$\text{rate} = \frac{\text{distance}}{\text{time}}$$

```
<m:r>
  <m:rPr>
    <m:nor/>
  </m:rPr>
  <m:t>rate</m:t>
</m:r>
<m:r>
  <m:t>=</m:t>
</m:r>
<m:f>
  <m:num>
    <m:r>
      <m:rPr>
        <m:nor/>
      </m:rPr>
      <m:t>distance</m:t>
    </m:r>
  </m:num>
```

```
    <m:den>
      <m:r>
        <m:rPr>
          <m:nor/>
        </m:rPr>
        <m:t>time</m:t>
      </m:r>
    </m:den>
  </m:f>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.75 num (Numerator)

This element specifies the numerator of the Fraction object f. [*Example*: The a in $^{a}/_{b}$:

```
  <m:f>
    <m:fPr>
      <m:type  m:val="skw"/>
    </m:fPr>
    <m:num>
      <m:r>
        <m:t>a</m:t>
      </m:r>
    </m:num>
    <m:den>
      <m:r>
        <m:t>b</m:t>
      </m:r>
    </m:den>
  </m:f>
```


end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathArg) is located in §A.6.1. *end note*]

22.1.2.76 objDist (Object Distribution)

This element specifies Array Object Distribution. When `1` or `true`, the contents of the array are spaced to the maximum width of the array object. When this element is omitted, the array does not receive object distribution. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

[*Example*:

$$\begin{aligned} 111a &= c \\ 111a + 111b &= 111c \\ b &= 111c \end{aligned}$$

is represented by the following XML:

```
<m:eqArr>
  <m:eqArrPr>
<m:maxDist m:val="1"/>
    <m:objDist m:val="1"/>
  </m:eqArrPr>
  <m:e>
    <m:r>
      <m:t>111&amp;a&amp;&amp;&amp;&amp;&amp;&amp;=&amp;&amp;c</m:t>
    </m:r>
  </m:e>
  <m:e>
    <m:r>
      <m:t>111&amp;a&amp;&amp;+&amp;111&amp;b&amp;&amp;=&amp;111&amp;c</m:t>
    </m:r>
  </m:e>
  <m:e>
    <m:r>
      <m:t>&amp;&amp;&amp;&amp;&amp;b&amp;&amp;=&amp;111&amp;c</m:t>
    </m:r>
  </m:e>
</m:eqArr>
```

Because object distribution and maximum distribution are applied, the equation array is expanded to fill the width that is allowed and the columns are distributed horizontally within the object to the bounds of the object.

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.77 oMath (Office Math)

This element specifies an instance of mathematical text. When used independently (not inside an oMathPara) with non-mathematical text preceding and/or following it, an independent oMath is interpreted as an inline math zone. All such math zones, including equations, expressions, arrays of equations or expressions, and formulas are represented by oMath blocks. When used in a display math zone (a math paragraph, oMathPara), oMath is a container for an instance of mathematical text that starts on its own line and is not an inline math zone. When an oMath block is part of a display math zone, it is not itself an inline math zone. When an oMath block is not part of a display math zone, it is interpreted as its own inline math zone. The contents of an oMath block do not differ between display zone containers and independent inline math zones.

[*Example*:

$1 + 1 = 2$

```
<m:oMath>
  <m:r>
    <m:t>1+1=2</m:t>
  </m:r>
</m:oMath>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMath) is located in §A.6.1. *end note*]

22.1.2.78 oMathPara (Office Math Paragraph)

This element specifies a math paragraph, or display math zone, that contains one or more oMath elements that are in display mode. The oMath containers of a display math zone are not themselves considered inline math zones..

[*Example*:

$1 + 1 = 2$
$2 + 2 = 4$ is represented by the following XML:

```
<m:oMathPara>
  <m:oMath>
    <m:r>
      <m:t>1+1=2</m:t>
    </m:r>
    <m:r>
      <w:br/>
    </m:r>
  </m:oMath>
  <m:oMath>
    <m:r>
      <m:t>2+2=4</m:t>
    </m:r>
  </m:oMath>
</m:oMathPara>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathPara) is located in §A.6.1. *end note*]

22.1.2.79 oMathParaPr (Office Math Paragraph Properties)

This property specifies properties of the math paragraph oMathPara, including justification jc.

[*Example*:

$1 + 1 = 2$
$2 + 2 = 4$ is represented by the following XML:

```
<m:oMathPara>
  <m:oMathParaPr>
    <m:jc m:val="right"/>
  </m:oMathParaPr>
  <m:oMath>
    <m:r>
```

```
        <m:t>1+1=2</m:t>
      </m:r>
      <m:r>
        <w:br/>
      </m:r>
    </m:oMath>
    <m:oMath>
      <m:r>
        <m:t>2+2=4</m:t>
      </m:r>
    </m:oMath>
  </m:oMathPara>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathParaPr) is located in §A.6.1. *end note*]

22.1.2.80 opEmu (Operator Emulator)

This element specifies the Operator Emulator property on box. When 1 or true, the box and its contents behave as a single operator and inherit the properties of an operator. This means, for example, that the character can serve as a point for a line break and can be aligned to other operators. (For more details on the properties of an operator, see Unicode Technical Report #25, §3.2.2 and §3.2.3, and Unicode Technical Note #28.) Operator Emulators are often used when one or more glyphs combine to form an operator, such as $==$.

When the element is absent, the default value of the property is 0 meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied.

[*Example*: The following equation uses an Operator Emulator:

$$a == b$$

Its XML representation is as follows:

```
<m:r>
  <m:t>a</m:t>
</m:r>
<m:box>
  <m:boxPr>
    <m:opEmu m:val="1"/>
    <m:aln/>
  </m:boxPr>
```



```
    <m:e>
      <m:r>
        <m:t>==</m:t>
      </m:r>
    </m:e>
  </m:box>
  <m:r>
    <m:t>b</m:t>
  </m:r>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.81 phant (Phantom Object)

This element specifies the phantom object. This object has two primary uses: adding the spacing of the phantom base element e without displaying that base; and suppressing part of the glyphfor spacing considerations. [*Example*: In the right column below, the descender of the y is suppressed (set to zero) while in the left column, it is shown. The presence or absence of the descender changes the shape of the $\sqrt{y}$.

Without `<m:phant>` $\sqrt{y}$	With `<m:phant>` $\sqrt{y}$

`<m:rad>`
`  <m:radPr>`
`  <m:degHide m:val="1" />`
`  </m:radPr>`
`  <m:deg />`
`  <m:e>`
`  <m:r>`
`  <m:t>y</m:t>`
`  </m:r>`
`  </m:e>`
	`<m:rad>`
	`  <m:radPr>`
	`  <m:degHide m:val="1" />`
	`  </m:radPr>`
	`  <m:deg />`
	`  <m:e>`
	`  <m:phant>`
	`  <m:phantPr>`
	`  <m:zeroDesc m:val="1" />`
	`  </m:phantPr>`

</m:rad>	<m:e> <m:r> <m:t>y</m:t> </m:r> </m:e> </m:phant> </m:e> </m:rad>

end example]

[*Note*: In typography, a *glyph* is the shape given in a particular typeface to a specific symbol. Most scripts share the notion of a baseline: an imaginary horizontal line on which characters rest. In some scripts, parts of glyphs lie below the baseline. The *descent* spans the distance between the baseline and the lowest descending glyph in a typeface, and the part of a glyph that descends below the baseline has the name *descender*. Conversely, the *ascent* spans the distance between the baseline and the top of the glyph that reaches farthest from the baseline. end note].

[*Note*: The W3C XML Schema definition of this element's content model (CT_Phant) is located in §A.6.1. *end note*]

22.1.2.82 phantPr (Phantom Properties)

This element specifies properties of the Phantom object, including whether the phantom is hidden or visible, and the amount of space that is taken into account when laying out text and objects around phantoms.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PhantPr) is located in §A.6.1. *end note*]

22.1.2.83 plcHide (Hide Placeholders (Matrix))

This element specifies the Hide Placeholders property on a matrix m. When this property is on, placeholders do not appear in the matrix. If this element is omitted, placeholders do appear such that the locations where text can be inserted are made visible. In other words, when the element is absent, the default value of the property is 0 meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied.

[*Example*: The following two examples of matrices show the hidden (right matrix) and visible (left matrix) states of placeholders. The key difference between the placeholders in these matrices is that the left matrix contains visual representations for each placeholder in the matrix while the right matrix does not. However, the contents of each matrix remain the same between these two examples:

$$\begin{pmatrix} 1 & \square & \square \\ \square & 1 & \square \\ \square & \square & 1 \end{pmatrix} \begin{pmatrix} 1 & & \\ & 1 & \\ & & 1 \end{pmatrix}$$

The plcHide element does not specify how a placeholder should be visually represented. It only specifies whether placeholders should or should not be visualized. The above example (left) is only one possible visual representation.

In the matrix described by the XML below, placeholders will be hidden:

```
<m:mPr>
  <m:plcHide  m:val="1"/>
  <m:mcs>
    <m:mc>
      <m:mcPr>
        <m:mcJc m:val="center"/>
        <m:count  m:val="3"/>
      </m:mcPr>
    </m:mc>
  </m:mcs>
</m:mPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.84 pos (Position)

This element specifies the position of the bar or group character in the parent object; the default is `bot`. Whether the element is absent or present without the val attribute, the default of the val attribute is `bot`.
[*Example*: The XML representation for the mathematical overbar is:

```
<m:bar>
  <m:barPr>
    <m:pos   m:val="top"/>
  </m:barPr>
  <m:e>
    <m:r>
      <m:t>a</m:t>
    </m:r>
  </m:e>
</m:bar>
```

end example]

Attributes	Description

val (Value)	Specifies the position of the parent element. Possible values are `top` and `bot`.
	
	[*Example*:
	`<m:barPr>`
	`  <m:pos  m:val="top"/>`
	`</m:barPr>`
	
	The possible values for this attribute are defined by the ST_TopBot simple type (§22.1.3.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TopBot) is located in §A.6.1. *end note*]

22.1.2.85 postSp (Post-Paragraph Spacing)

This element specifies the spacing after a math paragraph, in twips. If this element is omitted, no spacing is applied after the paragraph.

Attributes	Description
val (Value)	Specifies the value, in twips, of the parent element. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.6.1. *end note*]

22.1.2.86 preSp (Pre-Paragraph Spacing)

This element specifies the spacing before a math paragraph, in twips. If this element is omitted, no spacing is applied before the paragraph.

Attributes	Description
val (Value)	Specifies the value, in twips, of the parent element. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.6.1. *end note*]

22.1.2.87 r (Run)

This element specifies a run of mathematical text.

[*Example*:

$j + k$ is represented by the following XML:

```
<m:oMath>
  <m:e>
    <m:r>
      <m:t>j+k</m:t>
    </m:r>
  </m:e>
</m:oMath>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_R) is located in §A.6.1. *end note*]

22.1.2.88 rad (Radical Object)

This element specifies the radical object, consisting of a radical, a base e, and an optional degree deg. [*Example*: Examples of rad are $\sqrt[3]{x}$ (XML shown below) and $\sqrt{x}$.

```
<m:rad>
  <m:deg>
     <m:r>
        <m:rPr>
          <m:scr    m:val="roman"/>
          <m:sty    m:val="p"/>
        </m:rPr>
        <m:t>3</m:t>
     </m:r>
  </m:deg>
```

```
    <m:e>
      <m:r>
        <m:t>x</m:t>
      </m:r>
    </m:e>
  </m:rad>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rad) is located in §A.6.1. *end note*]

22.1.2.89 radPr (Radical Properties)

This element specifies properties of the Radical object rad, including the hidden or shown state of the degree deg. [*Example*:

$\sqrt{b}$ is represented by the following XML:

```
  <m:rad>
    <m:radPr>
      <m:degHide m:val="1"/>
    </m:radPr>
    <m:deg/>
    <m:e>
      <m:r>
        <m:t>b</m:t>
      </m:r>
    </m:e>
  </m:rad>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_RadPr) is located in §A.6.1. *end note*]

22.1.2.90 rMargin (Right Margin)

This element specifies the right margin for math, in twips. If this element is omitted, no right margin is used. In other words, when the element is absent, the default value of the option is 0. When the element is present and the val attribute is absent, the default of the val attribute is 1440 (or 1 inch).Math margins are added to the paragraph settings for margins. If the sum of lMargin and rMargin exceed the width available, lMargin should be ignored. If rMargin exceeds the width available, a default indent of 1440 twips should be used. [*Example*: The following XML demonstrates an rmargin setting of 1".


```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="1440"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

Attributes	Description
val (Value)	Specifies the value, in twips, of the parent element. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.6.1. *end note*]

22.1.2.91 rPr (Run Properties)

This element specifies the properties of the math run r.

[*Example*:

$\mathfrak{F} + \mathfrak{G}$ is represented by the following XML:

```
<m:r>
  <m:rPr>
    <m:scr m:val="fraktur"/>
    <m:sty m:val="b"/>
  </m:rPr>
  <m:t>F+G</m:t>
</m:r>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_RPR) is located in §A.6.1. *end note*]

22.1.2.92 rSp (Row Spacing (Array))

This element specifies spacing between rows of an array eqArr; it is used only when rSpRule is set to 3 (exactly; in which case the unit of measure is points) or 4 (Multiple; in which case the unit of measure is half-lines). If this element is omitted, single line spacing is used in the array, and no additional spacing is used in the layout of rows. Whether the element is absent or present without the val attribute, the default of the val attribute is 0.

[*Example*: Below are three examples of the same array, with single line spacing, 1.5 line spacing, and double line spacing:

$$\begin{pmatrix} x- & y+ & z= & 10 \\ 3x+ & y+2z= & & 34 \\ -5x+2y- & z= & & -14 \end{pmatrix} \begin{pmatrix} x- & y+ & z= & 10 \\ 3x+ & y+2z= & & 34 \\ -5x+2y- & z= & & -14 \end{pmatrix} \begin{pmatrix} x- & y+ & z= & 10 \\ 3x+ & y+2z= & & 34 \\ -5x+2y- & z= & & -14 \end{pmatrix}$$

The following eqArr $\begin{matrix} a=b+c \\ d+e=f \end{matrix}$ has rSp of 3 half-lines:

```
<m:eqArr>
  <m:eqArrPr>
    <m:rSpRule  m:val="4"/>
    <m:rSp m:val="3"/>
  </m:eqArrPr>
  <m:e>
    <m:r>
      <m:t>a=b+c</m:t>
    </m:r>
  </m:e>
  <m:e>
    <m:r>
      <m:t>d+e=f</m:t>
    </m:r>
  </m:e>
</m:eqArr>
```

end example]

Attributes	Description
val (Value)	Specifies the amount of space between columns of the parent element (for cGp/cSp) or rows (for rSp). The manner in which this value is determined depends on the setting of the rule of the parent element. For cGp, if the rule is set to 3 (or "Exactly"), then the unit is interpreted as twips. If the rule is set to 4 (or "Multiple"), then the unit is interpreted as number of 0.5 em increments.

Attributes	Description
	For cSp, this value is measured in twips. There is no corresponding cSpRule. For rSp, if the rule is set to 3 (or "Exactly"), then the unit is interpreted as points. If the rule is set to 4 (or "Multiple"), then the unit is interpreted as half-lines. The possible values for this attribute are defined by the ST_UnSignedInteger simple type (§22.1.3.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnSignedInteger) is located in §A.6.1. *end note*]

22.1.2.93 rSpRule (Row Spacing Rule)

This element specifies the type of vertical spacing between columns in a matrix. The following table demonstrates possible values of rSpRule along with their definitions and examples.

Value	Line spacing between rows	Example (non-normative)
0	Single line gap	1 2 3 4
1	1.5 line gap	1 2 3 4
2	2 line gap	1 2 3 4
3	Exactly (rely on value of rGp, measured in points)	1 2 3 4
4	Multiple (rely on value of rGp, measured in lines)	1 2 3 4

Whether the element is absent or present without the val attribute, the default of the val attribute is 0 (or single line gap).

[*Example*:

1 2
3 4

is represented by the following XML:

```
<m:m>
  <m:mPr>
    <m:rSpRule m:val="2"/>
```

```
    </m:mPr>
    <m:mr>
      <m:e>
        <m:r>
          <m:t>1</m:t>
        </m:r>
      </m:e>
      <m:e>
        <m:r>
          <m:t>2</m:t>
        </m:r>
      </m:e>
    </m:mr>
    <m:mr>
      <m:e>
        <m:r>
          <m:t>3</m:t>
        </m:r>
      </m:e>
      <m:e>
        <m:r>
          <m:t>4</m:t>
        </m:r>
      </m:e>
    </m:mr>
  </m:m>
```

Because the row spacing rule is set to 2, there is twice as much space between the matrix rows as usual.

end example]

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>val (Value)</td><td>Specifies the type of spacing between rows and/or columns. Possible values are 0, 1, 2, 3, or 4, whose definitions are contained in the following table:
<table>
<tr><th>Value</th><th>Column/Row Gap</th><th>Example</th></tr>
<tr><td>0</td><td>Single spacing gap (1 em)</td><td>1 2</td></tr>
<tr><td>1</td><td>1.5 spacing gap (1.5 ems)</td><td>1 2</td></tr>
<tr><td>2</td><td>2 spacing gap (2 ems)</td><td>1 2</td></tr>
<tr><td>3</td><td>Exactly
(for columns, rely on value of cGp, measured in twips)
(for rows, rely on value of</td><td>1 2</td></tr>
</table>
</td></tr>
</table>

Attributes	Description
	<table><tr><td></td><td>rSp, measured in points)</td><td></td></tr><tr><td>4</td><td>Multiple (for columns, rely on value of cGp, measured in 0.5 em increments) (for rows, rely on value of rSp, measured in lines)</td><td>1 2</td></tr></table> The possible values for this attribute are defined by the ST_SpacingRule simple type (§22.1.3.11).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SpacingRule) is located in §A.6.1. *end note*]

22.1.2.94 scr (Script)

This element describes the script applied to the characters in the run. The XML includes the Unicode value of the character (between U+0000 and U+007F), along with the script of the character. The application maps the value and script type to the appropriate Unicode range.

Whether the element is absent or present without the val attribute, the default of the val attribute is roman.

[*Example*: 𝔞𝕒

```
<m:r>
  <m:rPr>
    <m:scr m:val="fraktur"/>
    <m:sty m:val="p"/>
  </m:rPr>
  <m:t>a</m:t>
</m:r>
<m:r>
  <m:rPr>
    <m:scr m:val="double-struck"/>
    <m:sty m:val="p"/>
  </m:rPr>
  <m:t>a</m:t>
</m:r>
```

end example]

Attributes	Description
val (Value)	Specifies the script type of the parent element. Possible values are: `double-struck`, `fraktur`, `monospace`, `roman`, `sans-serif`, and `script`. The possible values for this attribute are defined by the ST_Script simple type (§22.1.3.9).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Script) is located in §A.6.1. *end note*]

22.1.2.95 sepChr (Delimiter Separator Character)

This element specifies the character that separates base arguments e in the delimiter object d. If this element is omitted, the default sepChr is '|'. In other words, when the element is absent, the default is for the delimiter object separator character to be U+2502 (BOX DRAWINGS LIGHT VERTICAL).

When the element is present and the val attribute is absent, the default of the val attribute is empty which means the delimiter object separator character is absent.

[*Example*: Examples of d, each with a different sepChr, are: $(a_1|a_2)(a_1:a_2)(a_1;a_2)$. The following example sets COLON (:) as the separator character:

```
<m:dPr>
  <m:sepChr val="&#0058;"/>
</m:dPr>
```

end example]

Attributes	Description

val (value)	Specifies the character used by the parent element. When it is omitted, the parent uses its assigned default.
	
	[*Example*: Accent object ã:
	
	`<m:accPr>`
	`  <m:chr m:val="̃"/>`
	`</m:accPr>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Char simple type (§22.1.3.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Char) is located in §A.6.1. *end note*]

22.1.2.96 show (Phantom Show)

This element specifies the show property of the phantom phant. When `0` or `false`, the phant base e is hidden. If this element is omitted, the base e is shown. Whether the element is absent or present without the val attribute, the default of the val attribute is `1` meaning that this property is applied.

[*Example*: In the following example, there is a phantom of the fraction a/b in the second radical such that only the height is preserved. The fraction does not show.

$$\sqrt{\frac{a}{b}} = \sqrt{x}$$

```
<m:phantPr>
  <m:show  m:val="0"/>
  <m:zeroDesc m:val="1"/>
</m:phantPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.97 shp (Shape (Delimiters))

This element specifies the shape of delimiters in the delimiter object d. Delimiters can be centered around the math axis of the mathematical text and still be made to fit the entire height of their contents (see right-hand example below), or their height and shape can be altered to exactly match their contents (see left-hand example below). These settings significantly impact the shape of the mathematical text. When this element is omitted, delimiters are '`centered`'. Whether the element is absent or present without the val attribute, the default of the val attribute is `centered`. [*Example*: In the examples below, delimiters will be matched to the exact shape of their contents contents on the left, and will be centered on the right:

Match Centered

$$a+b=\left(\frac{\frac{a}{\frac{b}{\frac{z}{r}}}}{c}+\frac{42}{z+y}\right)$$

```
<m:dPr>
  <m:shp  m:val="match"/>
</m:dPr>
```

$$a+b=\left(\frac{\frac{a}{\frac{b}{\frac{z}{r}}}}{c}+\frac{42}{z+y}\right)$$

```
<m:dPr>
  <m:shp  m:val="centered"/>
</m:dPr>
```

end example]

Attributes	Description
val (Value)	Specifies the shape of the parent element. Possible values are `match` and `centered`. The possible values for this attribute are defined by the ST_Shp simple type (§22.1.3.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Shp) is located in §A.6.1. *end note*]

22.1.2.98 smallFrac (Small Fraction)

This element specifies a reduced fraction size display math, such that the numerator and denominator are written in script size instead of at the size of regular text.

When the element is absent, the default value of the option is `0` meaning that this option is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this option is applied.

[*Example*: The XML containing this element in use is:

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```


end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.99 sPre (Pre-Sub-Superscript Object)

This element specifies the Pre-Sub-Superscript object, which consists of a base e and a subscript and superscript placed to the left of the base, as in ${}_{1}^{2}A$. [*Example*: The XML that specifies this function is:

```
<m:sPre>
  <m:sub>
    <m:r>
      <m:t>1</m:t>
    </m:r>
  </m:sub>
  <m:sup>
    <m:r>
      <m:t>2</m:t>
    </m:r>
  </m:sup>
  <m:e>
    <m:r>
      <m:t>A</m:t>
    </m:r>
  </m:e>
</m:sPre>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SPre) is located in §A.6.1. *end note*]

22.1.2.100 sPrePr (Pre-Sub-Superscript Properties)

This element specifies properties such as ctrlPr that can be stored on the Pre-Sub-Superscript object sPre.

[*Example*:

${}_{1}^{2}a$ is represented by the following XML:

```
<m:sPre>
  <m:sPrePr/>
  <m:sub>
    <m:r>
      <m:t>1</m:t>
    </m:r>
  </m:sub>
  <m:sup>
    <m:r>
      <m:t>2</m:t>
    </m:r>
  </m:sup>
  <m:e>
    <m:r>
      <m:t>a</m:t>
    </m:r>
  </m:e>
</m:sPre>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SPrePr) is located in §A.6.1. *end note*]

22.1.2.101 sSub (Subscript Object)

This element specifies the subscript object sSub, which consists of a base e and a reduced-size scr placed below and to the right, as in x_n. [*Example*: The XML that specifies this object is:

```
<m:sSub>
  <m:e>
    <m:r>
      <m:t>x</m:t>
    </r>
  </m:e>
  <m:sub>
    <m:r>
      <m:t>n</m:t>
    </r>
  </m:sub>
</m:sSub>
```


end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SSub) is located in §A.6.1. *end note*]

22.1.2.102 sSubPr (Subscript Properties)

This element specifies properties such as ctrlPr that can be stored on the Subscript object sSub.

[*Example*:

a_2 is represented by the following XML:

```
<m:sSub>
  <m:sSubPr/>
  <m:e>
    <m:r>
      <m:t>a</m:t>
    </m:r>
  </m:e>
  <m:sub>
    <m:r>
      <m:t>2</m:t>
    </m:r>
  </m:sub>
</m:sSub>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SSubPr) is located in §A.6.1. *end note*]

22.1.2.103 sSubSup (Sub-Superscript Object)

This element specifies the sub-superscript object, which consists of a base e, a reduced-size scr placed below and to the right, and a reduced-size scr placed above and to the right, as in x_m^n.

[*Example*: The XML that specifies this function is:

```
<m:sSubSup>
  <m:e>
    <m:r>
      <m:t>x</m:t>
    </r>
  </m:e>
```

```
    <m:sub>
      <m:r>
        <m:t>m</m:t>
      </r>
    </m:sub>
    <m:sup>
      <m:r>
        <m:t>n</m:t>
      </r>
    </m:sup>
  </m:sSubSup>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SSubSup) is located in §A.6.1. *end note*]

22.1.2.104 sSubSupPr (Sub-Superscript Properties)

This element specifies properties of the Sub-Superscript object, including the alignment of scripts.

[*Example*:

f_{100}^{200} is represented by the following XML:

```
  <m:sSubSup>
    <m:sSubSupPr>
      <m:alnScr m:val="1"/>
    </m:sSubSupPr>
    <m:e>
      <m:r>
        <m:t>f</m:t>
      </m:r>
    </m:e>
    <m:sub>
      <m:r>
        <m:t>100</m:t>
      </m:r>
    </m:sub>
    <m:sup>
      <m:r>
        <m:t>200</m:t>
      </m:r>
    </m:sup>
  </m:sSubSup>
```


end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SSubSupPr) is located in §A.6.1. *end note*]

22.1.2.105 sSup (Superscript Object)

This element specifies the superscript object sSup, which consists of a base e and a reduced-size scr placed above and to the right, as in x^n. [*Example*: The XML that specifies this object is:

```
<m:sSup>
  <m:e>
    <m:r>
      <m:t>x</m:t>
    </r>
  </m:e>
  <m:sup>
    <m:r>
      <m:t>n</m:t>
    </r>
  </m:sup>
</m:sSup>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SSup) is located in §A.6.1. *end note*]

22.1.2.106 sSupPr (Superscript Properties)

This element specifies properties such as ctrlPr that can be stored on the Superscript object sSup.

[*Example*:

a^2 is represented by the following XML:

```
<m:sSup>
  <m:sSupPr/>
  <m:e>
    <m:r>
      <m:t>a</m:t>
    </m:r>
  </m:e>
  <m:sup>
    <m:r>
      <m:t>2</m:t>
    </m:r>
  </m:sup>
```

</m:sSup>*end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SSupPr) is located in §A.6.1. *end note*]

22.1.2.107 strikeBLTR (Border Box Strikethrough Bottom-Left to Top-Right)

This element specifies the hidden or shown state of a strikethrough diagonal line from the bottom-left corner to the top-right corner of borderBox. When this element is omitted, the strikethrough is not drawn. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied. When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.When applied, a strikethrough is drawn, as in ~~abc~~. [*Example*:

```
<m:borderBox>
  <m:borderBoxPr>
    <m:hideTop  m:val="1"/>
    <m:hideBot  m:val="1"/>
    <m:hideLeft  m:val="1"/>
    <m:hideRight  m:val="1"/>
    <m:strikeBLTR m:val="1"/>
  </m:borderBoxPr>
  <m:e>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:borderBox>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.108 strikeH (Border Box Strikethrough Horizontal)

This element specifies the hidden or shown state of a strikethrough horizontal line in borderBox. When this element is omitted, the strikethrough is not drawn. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied. When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied. When on, a horizontal strikethrough is drawn, as in ~~*abc*~~. [*Example*:

```
<m:borderBox>
  <m:borderBoxPr>
    <m:hideTop  m:val="1"/>
    <m:hideBot  m:val="1"/>
    <m:hideLeft  m:val="1"/>
    <m:hideRight  m:val="1"/>
    <m:strikeH m:val="1"/>
  </m:borderBoxPr>
  <m:e>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:borderBox>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.109 strikeTLBR (Border Box Strikethrough Top-Left to Bottom-Right)

This element specifies the hidden or shown state of a strikethrough diagonal line from the top-left corner to the bottom-right corner of borderBox. When this element is omitted, the strikethrough is not drawn. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not

applied. When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied. When applied, a strikethrough is drawn, as in ~~abc~~. [*Example*:

```
<m:borderBox>
  <m:borderBoxPr>
    <m:hideTop  m:val="1"/>
    <m:hideBot  m:val="1"/>
    <m:hideLeft  m:val="1"/>
    <m:hideRight  m:val="1"/>
    <m:strikeTLBR m:val="1"/>
  </m:borderBoxPr>
  <m:e>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:borderBox>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.110 strikeV (Border Box Strikethrough Vertical)

This element specifies the hidden or shown state of a strikethrough vertical line in borderBox. When this element is omitted, the strikethrough is not drawn. In other words, when the element is absent, the default value of the property is 0 meaning that this property is not applied. When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied. When applied, a strikethrough is drawn, as in $\boxed{\phi}$. [*Example*:

```
<m:borderBox>
  <m:borderBoxPr>
    <m:strikeV m:val="1"/>
  </m:borderBoxPr>
  <m:e>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:borderBox>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.111 sty (style)

This element describes the script applied to the characters in the run. The XML includes the Unicode value of the character along with the style of the character. The application maps the value and style to the appropriate Unicode range. Whether the element is absent or present without the val attribute, the default of the val attribute is `i`.

[*Example*: $\mathbf{a}\boldsymbol{b}$

```
<m:oMath>
  <m:r>
    <m:rPr>
      <m:scr  m:val="roman"/>
      <m:sty m:val="b"/>
    </m:rPr>
    <m:t>ab</m:t>
  </m:r>
</m:oMath>
```

end example]

Attributes	Description
val (Value)	Specifies the style of the parent element. Possible values are `b` (bold), `i` (italic), `bi` (bold-italic), and `p` (plain). The possible values for this attribute are defined by the ST_Style simple type (§22.1.3.12).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Style) is located in §A.6.1. *end note*]

22.1.2.112 sub (Subscript (Pre-Sub-Superscript))

This element specifies the subscript of the Pre-Sub-Superscript object sPre. [*Example*: For example, the sub in the object ${}_{1}^{2}A$ is 1. An example of this element in use is:

```
<m:sPre>
   <m:sub>
    <m:r>
      <m:t>1</m:t>
    </m:r>
  </m:sub>
  <m:sup>
    <m:r>
      <m:t>2</m:t>
    </m:r>
  </m:sup>
  <m:e>
    <m:r>
      <m:t>A</m:t>
    </m:r>
  </m:e>
</m:sPre>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathArg) is located in §A.6.1. *end note*]

22.1.2.113 subHide (Hide Subscript (n-ary))

This element specifies the n-ary Hide Subscript property. When 1 or `true`, the lower limit does not appear, as in $\int^{\infty} \frac{x}{x+1} dx$. If this element is omitted, the lower limit appears.

When this element is omitted, the default is `false`. In other words, When the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

[*Example*: An example of this element in use is:

```
<m:naryPr>
  <m:chr  m:val="&#8747;"/>
  <m:subHide m:val="1"/>
</m:naryPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.114 sup (Superscript (Superscript object))

This element specifies the superscript of the superscript object sSup. For example, the sup in the superscript object x^n is n. [*Example*: An example of this element in use is:

```
<m:sSup>
  <m:e>
    <m:r>
      <m:t>x</m:t>
    </m:r>
  </m:e>
```

```
    <m:sup>
      <m:r>
        <m:t>n</m:t>
      </m:r>
    </m:sup>
  </m:sSup>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathArg) is located in §A.6.1. *end note*]

22.1.2.115 supHide (Hide Superscript (n-ary))

This element specifies the n-ary Hide Superscript property. When `1` or `true`, the upper limit does not appear, as in $\int_0 \frac{x}{x+1} dx$. If this element is omitted, the lower limit appears.

When this element is omitted, the default is `false`. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

[*Example*: An example of this element in use is:

```
<m:naryPr>
  <m:chr  m:val="&#8747;"/>
  <m:supHide m:val="1"/>
</m:naryPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.116 t (Text)

This element specifies the text in a math run r.

[*Example*:

$J + K + a + j$ is represented by the following XML:

```
<m:oMath>
  <m:e>
    <m:r>
      <m:t>J+K+a+j</m:t>
    </m:r>
  </m:e>
</m:oMath>
```

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.6.1. *end note*]

22.1.2.117 transp (Transparent (Phantom))

This element specifies that the phantom is transparent for spacing. This means that if the contents of the phantom are belonging to a special spacing class (such as binary operators, relational operators, differentials, etc.), the contents of that phantom are taken into consideration when laying out text. If transparency is turned off, then the contents of the phantom are ignored during layout. When this element is omitted, transparency is `0` or `false`. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

In the following example, transparency is not applied on the phantom around the differential term. $\int xdx$. The spacing is incorrect. In the following integral, the only difference is that transparency is applied: $\int x\,dx$. Now the spacing is correct.

[*Example*: An example of this element in XML is:

```
<m:phantPr>
  <m:zeroAsc  m:val="1"/>
  <m:zeroDesc  m:val="1"/>
  <m:transp  m:val="1"/>
</m:phantPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.118 type (Fraction type)

This element specifies the type of fraction f; the default is 'bar'. Whether the element is absent or present without the val attribute, the default of the val attribute is bar. Fractions types are:

Stacked Fraction: $\frac{a}{b}$

Skewed Fraction: a/b

Linear Fraction: a/b

Stack Object (No-Bar Fraction): $\binom{n}{k}$

[Example:

c/d is represented by the following XML:

```
<m:f>
```



```
    <m:fPr>
      <m:type m:val="skw"/>
    </m:fPr>
    <m:num>
      <m:r>
        <m:t>c</m:t>
      </m:r>
    </m:num>
    <m:den>
      <m:r>
        <m:t>d</m:t>
      </m:r>
    </m:den>
  </m:f>
```

end example]

Attributes	Description
val (Value)	Specifies the type of fraction. Possible values are `bar` (Bar Fraction), `lin` (Linear Fraction), `noBar` (No-Bar Fraction (Stack)), and `skw` (Skewed). The possible values for this attribute are defined by the ST_FType simple type (§22.1.3.4).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FType) is located in §A.6.1. *end note*]

22.1.2.119 vertJc (Vertical Justification)

This element, combined with pos of groupChrPr, specifies the vertical layout of the groupChr object. Where pos specifies the position of the grouping character, vertJc specifies the alignment of the object with respect to the baseline. For example, when the group character is above the object, vertJc of `top` signifies that the top of the object falls on the baseline; when vertJc is set to `bot`, the bottom of the object is on the baseline. The table below demonstrates the four possible combinations of groupChr layout:

Pos	vertJc	Layout
`top`	`top`	$a \underset{bcd}{\longleftarrow} e$
`top`	`bot`	$a \overbrace{bcd} e$
`bot`	`top`	$a \underbrace{bcd} e$
`bot`	`bot`	$a \overset{yields}{\longrightarrow} b$

When the element is absent, the default value of the property is top.

When the element is present and the val attribute is absent, the default of the val attribute is bot.

[*Example*:

$a + \underbrace{b+c}_{} + d$ is represented by the following XML:

```
<m:oMath>
  <m:r>
    <m:t>a+</m:t>
  </m:r>
  <m:groupChr>
    <m:groupChrPr>
      <m:vertJc m:val="bot"/>
    </m:groupChrPr>
    <m:e>
      <m:r>
        <m:t>b+c</m:t>
      </m:r>
    </m:e>
  </m:groupChr>
  <m:r>
    <m:t>+d</m:t>
  </m:r>
</m:oMath>
```

Since the vertical justification property is set to bot on the group-character object, the bottom of the group-character object is aligned with the baseline.

end example]

Attributes	Description
val (Value)	Specifies the position of the parent element. Possible values are top and bot. [*Example*: <m:barPr> <m:pos m:val="top"/> </m:barPr> The possible values for this attribute are defined by the ST_TopBot simple type (§22.1.3.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TopBot) is located in §A.6.1. *end note*]

22.1.2.120 wrapIndent (Wrap Indent)

This element specifies the indent of the wrapped line of an instance of mathematical text. The line or lines of a wrapped instance of mathematical text after the line break can either be indented by a specified amount from the left margin, or right aligned. The default indent is 1". In other words, whether the element is absent or present without the val attribute, the default of the val attribute is 1440 twips (or 1 inch).

[*Example*: The XML below demonstrates wrapIndent in use:

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

Attributes	Description
val (Value)	Specifies the value, in twips, of the parent element. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.6.1. *end note*]

22.1.2.121 wrapRight (Wrap Right)

This element specifies the right justification of the wrapped line of an instance of mathematical text. The line or lines of a wrapped instance of mathematical text after the line break can either be indented by a specified amount from the left margin, or right aligned. If this element is present, the continuation is right aligned.

When the element is absent, the default value of the option is 0 meaning that this option is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this option is applied.

[*Example*: An example of this element in use is:

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapRight/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.122 zeroAsc (Phantom Zero Ascent)

This element specifies that the phantom has zero ascent. The ascent of the contents of the phantom is not taken into account during layout. When this property is omitted, the phantom does have ascent (zero ascent is not applied).

When the element is absent, the default value of the property is 0 meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied.

In the following example, the differential term is contained in a phantom that zero ascent. As a result, spacing is reduced between the tip of the "d" and the radical bar: $\sqrt{xdx}$. [*Example*:

```
<m:phantPr>
  <m:zeroAsc  m:val="1"/>
</m:phantPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.123 zeroDesc (Phantom Zero Descent)

This element specifies that the phantom has zero descent. The descent of the contents of the phantom is not taken into account during layout. When this property is omitted, the phantom does have descent (zero descent is not applied).

When the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

[*Example*: In the following two examples, only the second has zero descent around the "y". Note that the radical is smaller than in the first case. $\sqrt{y}\sqrt{y}$

```
<m:phantPr>
  <m:zeroDesc  m:val="1"/>
</m:phantPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element.

Attributes	Description
	A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.124 zeroWid (Phantom Zero Width)

This element specifies that the phantom has zero width. The width of the contents of the phantom is not taken into account during layout. When this property is omitted, the phantom does have width (zero width is not applied).

When the element is absent, the default value of the property is 0 meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied.

[*Example*: In the folowing example, the second radical contains a phantom of the fraction a/b. The phantom has zero width, such that only the height grows to accommodate the hidden fraction: $\sqrt{\frac{a}{b}} = \sqrt{x}$

```
<m:phantPr>
  <m:show  m:val="0"/>
  <m:zeroDesc  m:val="1"/>
</m:phantPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.3 Simple Types

This is the complete list of simple types dedicated to Math.

22.1.3.1 ST_BreakBin (Break Binary Operators)

This defines how to represent binary operators with respect to a line-wrapping break. The line can wrap before the operator or after the operator; alternately, the operator can appear both at the end of the first line and the beginning of the second.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
after (After)	When line-wrapping breaks occur on binary operators, the binary operator appears after the break (at the start of the next line).
before (Before)	When line-wrapping breaks occur on binary operators, the binary operator appears before the break (at the end of the first line).
repeat (Repeat)	When line-wrapping breaks occur on binary operators, the binary operator appears on both sides of the break (at the end of the first line and the start of the next line).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BreakBin) is located in §A.6.1. *end note*]

22.1.3.2 ST_BreakBinSub (Break on Binary Subtraction)

This simple type specifies how to represent subtraction on both sides of a line-wrapping break, when the Break Binary Operators option is set to repeat. The first character represents the sign at the end of the line with the break; the second represents the sign at the start of the wrapped line. Options are --, -+, and +-.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
+- (Plus Minus)	Repetition of subtraction sign after a line-wrapping break is plus on the first line and minus on the second line.

Enumeration Value	Description
-+ (Minus Plus)	Repetition of subtraction sign after a line-wrapping break is minus on the first line and plus on the second line.
-- (Minus Minus)	Repetition of subtraction sign after a line-wrapping break is minus on the first and second lines.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BreakBinSub) is located in §A.6.1. *end note*]

22.1.3.3 ST_Char (Character)

This Simple Type specifies the single character used by the parent element.

[*Example*: In the following example, $\{a\}$ uses { and } as its enclosing characters, instead of the default (and).

```
<m:dPr>
  <m:begChr  m:val="{"/>
  <m:endChr m:val="}"/>
</m:dPr>
```

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 maximum length of 1 characters.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Char) is located in §A.6.1. *end note*]

22.1.3.4 ST_FType (Fraction Type)

Fractions can be of type bar (horizontal fraction bar), skewed ("skw" - diagonal fraction bar with kerned and vertically adjusted numerator and denominator), linear ("lin" - diagonal fraction bar, takes up exactly one line of space), and the "stack" object ("noBar").

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bar (Bar Fraction)	Fraction with a horizontal fraction bar.
lin (Linear Fraction)	Fraction with slanted fraction bar, that takes up no

Enumeration Value	Description
	additional vertical space.
noBar (No-Bar Fraction (Stack))	Stack object, which looks like a fraction with no fraction bar.
skw (Skewed)	Fraction with diagonal fraction bar.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FType) is located in §A.6.1. *end note*]

22.1.3.5 ST_Integer2 (Integer value (-2 to 2))

This simple type contains a value from (-2,+2) which specifies the size of the argument. The effects of each value are described by the referencing element.

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 -2.
- This simple type has a maximum value of less than or equal to 2.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Integer2) is located in §A.6.1. *end note*]

22.1.3.6 ST_Integer255 (Integer value (1 to 255))

This simple type specifies an integer value. The semantics of each value are discussed by the referencing element.

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 1.
- This simple type has a maximum value of less than or equal to 255.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Integer255) is located in §A.6.1. *end note*]

22.1.3.7 ST_Jc (Justification)

This Simple Type specifies the justification of Math Paragraphs. Justification of the Math Paragraph can be Left, Right, Centered, or Centered as Group.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
center (Center (Text))	Centers each instance of mathematical text individually with respect to margins.
centerGroup (Centered as Group (Text))	Justifies instances of mathematical text with respect to each other, and centers the group of mathematical text (the Math Paragraph) with respect to the page.
left (Left Justification)	Left justification of Math Paragraph
right (Right)	Right Justification of Math Paragraph

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Jc) is located in §A.6.1. *end note*]

22.1.3.8 ST_LimLoc (Limit Location)

Limits can be in one of two positions: Under-Over (`undOvr` - above and below the base), and Subscript-Superscript (`subSup` - positioned to the side of the base, in the position of subscripts and superscripts).

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
subSup (Subscript-Superscript location)	Limits placed to the side of the base, as opposed to directly over and under.
undOvr (Under-Over location)	Limits placed to the directly above and below the base, as opposed to on the side.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LimLoc) is located in §A.6.1. *end note*]

22.1.3.9 ST_Script (Script)

Script can be of type Roman, Script, Fraktur, Double-Struck, Sans-Serif, or Monospace.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
double-struck (double-struck)	Double-Struck Script Type
fraktur (Fraktur)	Fraktur Script Type
monospace (Monospace)	Monospace Script Type

Enumeration Value	Description
roman (Roman)	Roman Script Type
sans-serif (Sans-Serif)	Sans-Serif Script Type
script (Script)	Script Type

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Script) is located in §A.6.1. *end note*]

22.1.3.10 ST_Shp (Shape (Delimiters))

Delimiters shape can be centered around the argument, or matched to the shape of the argument.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
centered (Centered (Delimiters))	Delimiters are centered around their argument.
match (Match)	Match shape of contents of delimiters.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Shp) is located in §A.6.1. *end note*]

22.1.3.11 ST_SpacingRule (Spacing Rule)

Integer value (0 to 4), representing the type of spacing between rows.

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 4.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SpacingRule) is located in §A.6.1. *end note*]

22.1.3.12 ST_Style (Style)

Style of math can be plain, bold, italic, or bold-italic (p, bi, i, or bi).

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bold)	Bold
bi (Bold-Italic)	Bold-Italic
i (Italic)	Italic
p (Plain)	Plain

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Style) is located in §A.6.1. *end note*]

22.1.3.13 ST_TopBot (Top-Bottom)

Possible values are `top` and `bot`.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bot (Bottom Alignment)	Aligns the bottom of the object to the baseline of the surrounding text.
top (Top)	Aligns the top row of the object to the baseline of the surrounding text.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TopBot) is located in §A.6.1. *end note*]

22.1.3.14 ST_UnSignedInteger (Unsigned integer.)

Unsigned Integer

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_UnSignedInteger) is located in §A.6.1. *end note*]

22.2 Extended Properties

Extended properties are a predefined set of metadata properties that are applicable to Office Open XML documents. These properties extend the set of core properties defined in Part 2: "Open Packaging Conventions" which are common to all packages.

Extended properties are stored within an Extended File Properties part with:

- Source Relationship: http://purl.oclc.org/ooxml/officeDocument/relationships/extendedProperties

- Content Type: application/vnd.openxmlformats-officedocument.extended-properties+xml

Each extended property is represented as an element in the extended properties part. Extended properties elements are non-repeatable and can be empty or omitted. If all extended property elements are omitted then the extended properties part can be excluded from a document.

[*Example*: A sample extended file properties part:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Properties xmlns="http://…/extended-properties"
  xmlns:vt="http://…/docPropsVTypes">
..<Template>Sales Report.dotm</Template>
..<TotalTime>0</TotalTime>
..<Pages>1</Pages>
..<Words>166</Words>
..<Characters>948</Characters>
..<Application>Microsoft Office Word</Application>
..<DocSecurity>0</DocSecurity>
..<Lines>7</Lines>
..<Paragraphs>2</Paragraphs>
..<ScaleCrop>false</ScaleCrop>
..<Company>Northwind Traders</Company>
..<LinksUpToDate>false</LinksUpToDate>
..<CharactersWithSpaces>1112</CharactersWithSpaces>
..<SharedDoc>false</SharedDoc>
..<HyperlinksChanged>false</HyperlinksChanged>
..<AppVersion>12.0000</AppVersion>
</Properties>
```

end example]

22.2.1 Table of Contents

This subclause is informative.

End of informative text.

22.2.2 Elements

The following elements specify the contents of this namespace:

22.2.2.1 Application (Application Name)

This element specifies the name of the application that created this document.

The possible values for this element are defined by the W3C XML Schema string datatype.

22.2.2.2 AppVersion (Application Version)

This element specifies the version of the application which produced this document.

The content of this element shall be of the form XX.YYYY where X and Y represent numerical values, or the document shall be considered non-conformant.

[*Note*: The contents of this element do not represent absolute values, but rather qualify the contents of the Application element to differentiate between different versions of the same producer. Applications should use this information in an informative manner only (as document metadata). *end note*]

The possible values for this element are defined by the W3C XML Schema string datatype.

22.2.2.3 Characters (Total Number of Characters)

This element specifies the total number of characters in a document.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.4 CharactersWithSpaces (Number of Characters (With Spaces))

This element specifies the last count of the number of characters (including spaces) in this document.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.5 Company (Name of Company)

This element specifies the name of a company associated with the document.

The possible values for this element are defined by the W3C XML Schema string datatype.

22.2.2.6 DigSig (Digital Signature)

This element contains the signature of a digitally signed document.

[*Note*: This property is a mechanism used by legacy documents to store the digital signature of its binary representation, and should be avoided in favor of the well-defined mechanism defined in Part 2. Any use of this property should be for legacy compatibility only, and is application-defined. *end note*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DigSigBlob) is located in §A.6.2. *end note*]

22.2.2.7 DocSecurity (Document Security)

This metadata element specifies the security level of a document as a numeric value. Document security is defined as:

DocSecurity	Security Level
1	Document is password protected.
2	Document is recommended to be opened as read-only.
4	Document is enforced to be opened as read-only.
8	Document is locked for annotation.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.8 HeadingPairs (Heading Pairs)

Heading pairs indicates the grouping of document parts and the number of parts in each group. These parts are not document parts but conceptual representations of document sections.

[*Example*: A presentation composing of three slides with an applied theme can have the following HeadingPairs representation:

```
<HeadingPairs>
  <vt:vector size="4" baseType="variant">
    <vt:variant>
      <vt:lpstr>Theme</vt:lpstr>
    </vt:variant>
    <vt:variant>
      <vt:i4>1</vt:i4>
    </vt:variant>
    <vt:variant>
      <vt:lpstr>Slide Titles</vt:lpstr>
    </vt:variant>
    <vt:variant>
      <vt:i4>3</vt:i4>
    </vt:variant>
  </vt:vector>
</HeadingPairs>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_VectorVariant) is located in §A.6.2. *end note*]

22.2.2.9 HiddenSlides (Number of Hidden Slides)

This element specifies the number of hidden slides in a presentation document.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.10 HLinks (Hyperlink List)

This element specifies the set of hyperlinks that were in this document when last saved.

[*Note*: The W3C XML Schema definition of this element's content model (CT_VectorVariant) is located in §A.6.2. *end note*]

22.2.2.11 HyperlinkBase (Relative Hyperlink Base)

This element specifies the base string used for evaluating relative hyperlinks in this document.

The possible values for this element are defined by the W3C XML Schema string datatype.

22.2.2.12 HyperlinksChanged (Hyperlinks Changed)

This element specifies that one or more hyperlinks in this part were updated exclusively in this part by a producer. The next producer to open this document shall update the hyperlink relationships with the new hyperlinks specified in this part.

The possible values for this element are defined by the W3C XML Schema boolean datatype.

22.2.2.13 Lines (Number of Lines)

This element specifies the total number of lines in a document when last saved by a conforming producer if applicable.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.14 LinksUpToDate (Links Up-to-Date)

This element indicates whether hyperlinks in a document are up-to-date. Set this element to TRUE to indicate that hyperlinks are updated. Set this element to FALSE to indicate that hyperlinks are outdated.

The possible values for this element are defined by the W3C XML Schema boolean datatype.

22.2.2.15 Manager (Name of Manager)

This element specifies the name of a supervisor associated with the document.

The possible values for this element are defined by the W3C XML Schema string datatype.

22.2.2.16 MMClips (Total Number of Multimedia Clips)

This element specifies the total number of sound or video clips that are present in the document.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.17 Notes (Number of Slides Containing Notes)

This element specifies the number of slides in a presentation containing notes.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.18 Pages (Total Number of Pages)

This element specifies the total number of pages of a document if applicable.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.19 Paragraphs (Total Number of Paragraphs)

This element specifies the total number of paragraphs found in a document if applicable.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.20 PresentationFormat (Intended Format of Presentation)

This element specifies the intended format for a presentation document. For example, a presentation intended to be shown on video has PresentationFormat "Video".

The possible values for this element are defined by the W3C XML Schema string datatype.

22.2.2.21 Properties (Application Specific File Properties)

This element specifies the application properties of a document. For properties of type string, NCR escape format (_xHHHH_) is used for any invalid XML characters.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Properties) is located in §A.6.2. *end note*]

22.2.2.22 ScaleCrop (Thumbnail Display Mode)

This element indicates the display mode of the document thumbnail. Set this element to TRUE to enable scaling of the document thumbnail to the display. Set this element to FALSE to enable cropping of the document thumbnail to show only sections that fits the display.

The possible values for this element are defined by the W3C XML Schema boolean datatype.

22.2.2.23 SharedDoc (Shared Document)

This element indicates if this document is currently shared between multiple producers. If this element is set to TRUE, producers should take care when updating the document.

The possible values for this element are defined by the W3C XML Schema boolean datatype.

22.2.2.24 Slides (Slides Metadata Element)

This element specifies the total number of slides in a presentation document.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.25 Template (Name of Document Template)

This element specifies the name of an external document template containing format and style information used to create the current document.

The possible values for this element are defined by the W3C XML Schema string datatype.

22.2.2.26 TitlesOfParts (Part Titles)

This element specifies the title of each document. These parts are not document parts but conceptual representations of document sections.

[*Example*: A presentation composing of three slides with an applied theme "Currency" can have the following TitlesofParts representation:


```
<TitlesofParts>
  <vt:vector size="4" baseType="lpstr">
    <vt:lpstr>Currency</vt:lpstr>
    <vt:lpstr>Slide 1</vt:lpstr>
    <vt:lpstr>Slide 2</vt:lpstr>
    <vt:lpstr>Slide 3</vt:lpstr>
  </vt:vector>
</TitlesofParts>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_VectorLpstr) is located in §A.6.2. *end note*]

22.2.2.27 TotalTime (Total Edit Time Metadata Element)

Total time that a document has been edited. The default time unit is minutes.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.28 Words (Word Count)

This element specifies the total number of words contained in a document when last saved.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.3 Custom Properties

Custom properties enable users to define custom metadata properties through a set of well-defined data types.

Custom properties are represented by property elements (§22.3.2.2) stored in the Custom File Properties part with:

- Source Relationship: http://purl.oclc.org/ooxml/officeDocument/relationships/customProperties
- Content Type: application/vnd.openxmlformats-officedocument.custom-properties+xml

Custom property elements are non-repeatable and can be empty or omitted. If all custom property elements are omitted then the custom properties part can be excluded from a document.

The type and value of custom properties are specified by child XML elements in the File Properties Variant Types namespace (discussed in detail in §22.4). User defined properties are uniquely identified through the name attribute of the property element. Custom properties can be associated with OLE document properties through the fmtid and pid attributes.

[*Example*: A custom OLE `Editor` property of type `string` can be defined as follows:

```
<property fmtid="{D5CDD505-2E9C-101B-9397-08002B2CF9AE}" pid="2"
  name="Editor">
  <vt:lpwstr>John Smith</vt:lpwstr>
</property>
```

end example]

22.3.1 Table of Contents

This subclause is informative.

End of informative text.

22.3.2 Elements

This subclause specifies the set of elements that define this namespace:

22.3.2.1 Properties (Custom File Properties)

Parent element for the custom file properties part.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Properties) is located in §A.6.3. *end note*]

22.3.2.2 property (Custom File Property)

This element specifies a single custom file property. Custom file property type is defined through child elements in the File Properties Variant Type namespace. Custom file property value can be set by setting the appropriate Variant Type child element value.

Attributes	Description
fmtid (Format ID)	Uniquely relates a custom property with an OLE property. The value of this attribute is a Globally Unique Identifier in the form of {HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHH} where each H is a hexidecimal. The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).
linkTarget (Bookmark Link Target)	Specifies the name of a bookmark in the current document (for WordprocessingML), or a table or named cell (for SpreadsheetML) from which the value of this custom document property should be extracted. If this attribute is present, then any value under this element shall be considered a cache and replaced with the value of this bookmark (if present) on save. If the bookmark is not present, then this link shall be considered broken and the cached value shall be retained.

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.
name (Custom File Property Name)	Specifies the name of this custom file property. The possible values for this attribute are defined by the W3C XML Schema string datatype.
pid (Property ID)	Uniquely relates a custom property with an OLE property. 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_Property) is located in §A.6.3. *end note*]

22.4 Variant Types

This subclause specifies the set of data types which can be included within file properties that accept variant type structures.

22.4.1 Table of Contents

This subclause is informative.

End of informative text.

22.4.2 Elements

The following elements define the contents of this namespace:

22.4.2.1 array (Array)

The array element defines the array variant type. Array contents shall be of uniform type as specified by the baseType attribute. The contents of an array are defined using repeated child elements of the appropriate variant type.

Multi-dimensional arrays can be defined by specifying the length of each dimension in the lBound and uBound attributes through the use of the "," delimiter. Child elements of multi-dimensional arrays are indexed along each dimension in the order the dimensions are declared.

In other words, the array shall be filled as follows:

- The first index shall be incremented to its maximum value [*Example*: [0,0,0] to [max,0,0] *end example*]
- The second index shall be incremented to its maximum value [*Example*: [0,1,0] to [0,max,0] *end example*]
- Subsequent indices shall be filled until all provided values have been added
- All other values shall have null values within the array (i.e. no default value shall be assumed).

[*Example*: A 2x3 variant type array of type "i4" is specified as follows:


```
<vt:array lBounds="0,0" uBounds="1,2" baseType="i4">
  <vt:i4>0</vt:i4>
  <vt:i4>1</vt:i4>
  <vt:i4>2</vt:i4>
  <vt:i4>3</vt:i4>
  <vt:i4>4</vt:i4>
</vt:array>
```

The resulting array: [0,0] = 0, [1,0] = 1, [0,1] = 2, [1,1] = 3, [0,2] = 4. *end example*]

Attributes	Description
baseType (Array Base Type)	The baseType attribute specifies the base variant type of an array. The allowed values are: variant, i1, i2, i4, int, ui1, ui2, ui4, uint, r4, r8, decimal, bstr, date, bool, cy, and error. The possible values for this attribute are defined by the ST_ArrayBaseType simple type (§22.4.3.1).
lBounds (Array Lower Bounds Attribute)	The lBounds attribute specifies the lower bound of an array in the format: #, #, # … # where each # represents an integer. The possible values for this attribute are defined by the W3C XML Schema int datatype.
uBounds (Array Upper Bounds Attribute)	The uBounds attribute specifies the upper bound of an array in the format: #, #, # … # where each # represents an integer. 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_Array) is located in §A.6.4. *end note*]

22.4.2.2 blob (Binary Blob)

This element specifies a base64 binary blob variant type.

This type is defined as follows: a DWORD count of bytes, followed by that many bytes of data. The byte count does not include the four bytes for the length of the count itself; an empty blob member would have a count of zero, followed by zero bytes.

The possible values for this element are defined by the W3C XML Schema base64Binary datatype.

22.4.2.3 bool (Boolean)

This element specifies a Boolean variant type.

The possible values for this element are defined by the W3C XML Schema boolean datatype.

22.4.2.4 bstr (Basic String)

This element defines a binary basic string variant type, which can store any valid Unicode character. Unicode characters that cannot be directly represented in XML as defined by the XML 1.0 specification, shall be escaped using the Unicode numerical character representation escape character format _xHHHH_, where H represents a hexadecimal character in the character's value. [*Example*: The Unicode character 8 is not permitted in an XML 1.0 document, so it shall be escaped as _x0008_. *end example*] To store the literal form of an escape sequence, the initial underscore shall itself be escaped (i.e. stored as _x005F_). [*Example*: The string literal `_x0008_` would be stored as `_x005F_x0008_`. *end example*]

The possible values for this element are defined by the W3C XML Schema string datatype.

22.4.2.5 clsid (Class ID)

This element specifies a class ID variant type. The value shall be a Globally Unique Identifier with format: {HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHH}.

The possible values for this element are defined by the ST_Guid simple type (§22.9.2.4).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Guid) is located in §A.6.9. *end note*]

22.4.2.6 cy (Currency)

This element specifies a currency variant type with exactly four digits after the decimal point.

The possible values for this element are defined by the ST_Cy simple type (§22.4.3.2).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Cy) is located in §A.6.4. *end note*]

22.4.2.7 date (Date and Time)

This element specifies a date variant type of type date-time as defined in RFC 3339.

The possible values for this element are defined by the W3C XML Schema dateTime datatype.

22.4.2.8 decimal (Decimal)

This element specifies a decimal variant type.

The possible values for this element are defined by the W3C XML Schema decimal datatype.

22.4.2.9 empty (Empty)

This element specifies an empty variant type. No values or child elements are allowed.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.6.4. *end note*]

22.4.2.10 error (Error Status Code)

The error element specifies a 32-bit error status code variant type of the form 0xHHHHHHHH. Each H represents a hexadecimal digit.

The possible values for this element are defined by the ST_Error simple type (§22.4.3.3).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Error) is located in §A.6.4. *end note*]

22.4.2.11 filetime (File Time)

This element specifies a file-time variant type of type date-time as defined in RFC 3339.

The possible values for this element are defined by the W3C XML Schema dateTime datatype.

22.4.2.12 i1 (1-Byte Signed Integer)

This element specifies a 1-byte signed integer variant type.

The possible values for this element are defined by the W3C XML Schema byte datatype.

22.4.2.13 i2 (2-Byte Signed Integer)

This element specifies a 2-byte signed integer variant type.

The possible values for this element are defined by the W3C XML Schema short datatype.

22.4.2.14 i4 (4-Byte Signed Integer)

This element specifies a 4-byte signed integer variant type.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.4.2.15 i8 (8-Byte Signed Integer)

This element specifies a 8-byte signed integer variant type.

The possible values for this element are defined by the W3C XML Schema long datatype.

22.4.2.16 int (Integer)

This element specifies an integer variant type.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.4.2.17 lpstr (LPSTR)

This element specifies a string variant type. For all characters that cannot be represented in XML as defined by the XML 1.0 specification, the characters are escaped using the Unicode numerical character representation escape character format _xHHHH_, where H represents a hexadecimal character in the character's value.

[*Example*: The Unicode character 8 is not permitted in an XML 1.0 document, so it must be escaped as _x0008_. *end example*]

The possible values for this element are defined by the W3C XML Schema string datatype.

22.4.2.18 lpwstr (LPWSTR)

This element specifies a string variant type. For all characters that cannot be represented in XML as defined by the XML 1.0 specification, the characters are escaped using the Unicode numerical character representation escape character format _xHHHH_, where H represents a hexadecimal character in the character's value. [*Example*: The Unicode character 8 is not permitted in an XML 1.0 document, so it must be escaped as _x0008_. *end example*]

The possible values for this element are defined by the W3C XML Schema string datatype.

22.4.2.19 null (Null)

This element specifies a null variant type.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Null) is located in §A.6.4. *end note*]

22.4.2.20 oblob (Binary Blob Object)

This element specifies a base64 binary blob object variant type.

This type is defined as follows: A blob member that contains a serialized object in the same representation that would appear in the ostream element. That is, a DWORD byte count (where the byte count does not include the size of itself) which is in the format of a class identifier followed by initialization data for that class.

The possible values for this element are defined by the W3C XML Schema base64Binary datatype.

22.4.2.21 ostorage (Binary Storage Object)

This element specifies a base64 binary storage object variant type.

This type is defined as follows: Identical to the storage element, but indicates that the designated storage shall contain a loadable object.

The possible values for this element are defined by the W3C XML Schema base64Binary datatype.

22.4.2.22 ostream (Binary Stream Object)

This element specifies a binary stream object variant type.

This type is defined as follows: Identical to the definition of the stream element, but indicates that the stream contains a serialized object, which is a CLSID – see the ST_Guid simple type (§22.9.2.4) – followed by initialization data for the specified class.

The possible values for this element are defined by the W3C XML Schema base64Binary datatype.

22.4.2.23 r4 (4-Byte Real Number)

This element specifies a 4-byte real number variant type.

The possible values for this element are defined by the W3C XML Schema float datatype.

22.4.2.24 r8 (8-Byte Real Number)

This element specifies a 8-byte real number variant type.

The possible values for this element are defined by the W3C XML Schema double datatype.

22.4.2.25 storage (Binary Storage)

This element specifies a binary storage variant type.

This type is defined as follows: Contains the base64-encoded data for a structured storage.

The possible values for this element are defined by the W3C XML Schema base64Binary datatype.

22.4.2.26 stream (Binary Stream)

This element specifies a binary stream variant type.

This type is defined as follows: Contains the base64-encoded data for a structured storage stream.

The possible values for this element are defined by the W3C XML Schema base64Binary datatype.

22.4.2.27 ui1 (1-Byte Unsigned Integer)

This element specifies a 1-byte unsigned integer variant type.

The possible values for this element are defined by the W3C XML Schema unsignedByte datatype.

22.4.2.28 ui2 (2-Byte Unsigned Integer)

This element specifies a 2-byte unsigned integer variant type.

The possible values for this element are defined by the W3C XML Schema unsignedShort datatype.

22.4.2.29 ui4 (4-Byte Unsigned Integer)

This element specifies a 4-byte unsigned integer variant type.

The possible values for this element are defined by the W3C XML Schema unsignedInt datatype.

22.4.2.30 ui8 (8-Byte Unsigned Integer)

This element specifies a 8-byte unsigned integer variant type.

The possible values for this element are defined by the W3C XML Schema unsignedLong datatype.

22.4.2.31 uint (Unsigned Integer)

This element specifies an unsigned integer variant type.

The possible values for this element are defined by the W3C XML Schema unsignedInt datatype.

22.4.2.32 variant (Variant)

This element can contain exactly 1 child element of any variant type. This element is only valid as a child element of a vector or array variant type.

[*Example*: A vector of variant types:

```
<vt:vector baseType="variant">
  <vt:variant>
    <vt:i4>12</vt:i4>
  </vt:variant>
  <vt:variant>
    <vt:lpstr>WorkSheets</vt:lpstr>
  </vt:variant>
</vt:vector>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Variant) is located in §A.6.4. *end note*]

22.4.2.33 vector (Vector)

This element defines the vector variant type. Vector contents shall be of uniform type as specified by the baseType attribute. The contents of a vector are defined using repeated child elements of the appropriate variant type.

[*Example*: A vector of lpstr variant types:

```
  <vt:vector baseType="lpstr">
    <vt:lpstr>One</vt:lpstr>
    <vt:lpstr>Two</vt:lpstr>
    <vt:lpstr>Three</vt:lpstr>
  </vt:vector>
```

end example]

Attributes	Description
baseType (Vector Base Type)	The baseType attribute specifies the base variant type of a vector. The allowed values are: variant, i1, i2, i4, i8, ui1, ui2, ui4, ui8, r4, r8, lpstr, lpwstr, bstr, date, filetime, bool, cy, error, and clsid.

Attributes	Description
	The possible values for this attribute are defined by the ST_VectorBaseType simple type (§22.4.3.4).
size (Vector Size)	Specifies the number of elements in the vector. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Vector) is located in §A.6.4. *end note*]

22.4.2.34 vstream (Binary Versioned Stream)

This element specifies a binary versioned stream variant type.

This type is defined as follows: A stream element's content with a GUID version (the version attribute).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Vstream) is located in §A.6.4. *end note*]

22.4.3 Simple Types

This is the complete list of simple types dedicated to Variant Types.

22.4.3.1 ST_ArrayBaseType (Array Base Type Simple Type)

The ST_ArrayBaseType simple type defines the allowed values for an array's baseType attribute as: variant, i1, i2, i4, int, ui1,ui2, ui4, uint, r4, r8, decimal, bstr, date, bool, cy, and error.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bool (Boolean Base Type)	Specifies that the variant type for the contents of a array shall be `bool`.
bstr (Basic String Base Type)	Specifies that the variant type for the contents of a array shall be `bstr`.
cy (Curency Base Type)	Specifies that the variant type for the contents of a array shall be `cy`.
date (Date and Time Base Type)	Specifies that the variant type for the contents of a array shall be `date`.
decimal (Decimal Base Type)	Specifies that the variant type for the contents of a array shall be `decimal`.
error (Error Status Code Base Type)	Specifies that the variant type for the contents of a

Enumeration Value	Description
	array shall be `error`.
i1 (1-Byte Signed Integer Base Type)	Specifies that the variant type for the contents of a array shall be `i1`.
i2 (2-Byte Signed Integer Base Type)	Specifies that the variant type for the contents of a array shall be `i2`.
i4 (4-Byte Signed Integer Base Type)	Specifies that the variant type for the contents of a array shall be `i4`.
int (Integer Base Type)	Specifies that the variant type for the contents of a array shall be `int`.
r4 (4-Byte Real Number Base Type)	Specifies that the variant type for the contents of a array shall be `r4`.
r8 (8-Byte Real Number Base Type)	Specifies that the variant type for the contents of a array shall be `r8`.
ui1 (1-Byte Unsigned Integer Base Type)	Specifies that the variant type for the contents of a array shall be `ui1`.
ui2 (2-Byte Unsigned Integer Base Type)	Specifies that the variant type for the contents of a array shall be `ui2`.
ui4 (4-Byte Unsigned Integer Base Type)	Specifies that the variant type for the contents of a array shall be `ui4`.
uint (Unsigned Integer Base Type)	Specifies that the variant type for the contents of a array shall be `uint`.
variant (Variant Base Type)	Specifies that the variant type for the contents of a array shall be `variant`.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ArrayBaseType) is located in §A.6.4. *end note*]

22.4.3.2 ST_Cy (Currency Simple Type)

The ST_Cy simple type defines the cy element as a currency variant type with exactly four digits after the decimal point.

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 shall match the following regular expression pattern: \s*[0-9]*\.[0-9]{4}\s*.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Cy) is located in §A.6.4. *end note*]

22.4.3.3 ST_Error (Error Status Code Simple Type)

The ST_Error simple type defines a 32-bit error status code variant type of the form 0xHHHHHHHH. Each H represents a hexadecimal.

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 shall match the following regular expression pattern: \s*0x[0-9A-Za-z]{8}\s*.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Error) is located in §A.6.4. *end note*]

22.4.3.4 ST_VectorBaseType (Vector Base Type Simple Type)

The ST_VectorBaseType simple type defines the allowed values for a vector's baseType attribute as: variant, i1, i2, i4, i8, ui1, ui2, ui4, ui8, r4, r8, lpstr, lpwstr, bstr, date, filetime, bool, cy, error, and clsid.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bool (Boolean Base Type)	Specifies that the variant type for the contents of a vector shall be `bool`.
bstr (Basic String Base Type)	Specifies that the variant type for the contents of a vector shall be `bstr`.
clsid (Class ID Base Type)	Specifies that the variant type for the contents of a vector shall be `clsid`.
cy (Currency Base Type)	Specifies that the variant type for the contents of a vector shall be `cy`.
date (Date and Time Base Type)	Specifies that the variant type for the contents of a vector shall be `date`.
error (Error Status Code Base Type)	Specifies that the variant type for the contents of a vector shall be `error`.
filetime (File Time Base Type)	Specifies that the variant type for the contents of a vector shall be `filetime`.
i1 (Vector Base Type Enumeration Value)	Specifies that the variant type for the contents of a vector shall be `i1`.
i2 (2-Byte Signed Integer Base Type)	Specifies that the variant type for the contents of a vector shall be `i2`.
i4 (4-Byte Signed Integer Base Type)	Specifies that the variant type for the contents of a vector shall be `i4`.

Enumeration Value	Description
i8 (8-Byte Signed Integer Base Type)	Specifies that the variant type for the contents of a vector shall be `i8`.
lpstr (LPSTR Base Type)	Specifies that the variant type for the contents of a vector shall be `lpstr`.
lpwstr (LPWSTR Base Type)	Specifies that the variant type for the contents of a vector shall be `lpwstr`.
r4 (4-Byte Real Number Base Type)	Specifies that the variant type for the contents of a vector shall be `r4`.
r8 (8-Byte Real Number Base Type)	Specifies that the variant type for the contents of a vector shall be `r8`.
ui1 (1-Byte Unsigned Integer Base Type)	Specifies that the variant type for the contents of a vector shall be `ui1`.
ui2 (2-Byte Unisigned Integer Base Type)	Specifies that the variant type for the contents of a vector shall be `ui2`.
ui4 (4-Byte Unsigned Integer Base Type)	Specifies that the variant type for the contents of a vector shall be `ui4`.
ui8 (8-Byte Unsigned Integer Base Type)	Specifies that the variant type for the contents of a vector shall be `ui8`.
variant (Variant Base Type)	Specifies that the variant type for the contents of a vector shall be `variant`.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_VectorBaseType) is located in §A.6.4. *end note*]

22.5 Custom XML Data Properties

This namespace defines the set of properties that can be associated with one or more custom XML parts within an Office Open XML document. A *custom XML part* is a part within an Office Open XML document, that contains arbitrary custom XML markup not necessarily defined by ISO/IEC 29500, and which is kept independent from the presentation-specific markup within the package.

[*Rationale*: It is often necessary to include custom XML semantics with an Office Open XML document, to store a complex set of properties (e.g., a document management system's metadata) along with the file. This mechanism allows this custom XML to be stored in the document in a way that is independent of the type of document and separate from the presentation markup—only the custom XML is stored in this part. *end rationale*]

The properties that can be applied to a custom XML part are:

- A part ID
- (optionally) One or more associated custom XML schemas

22.5.1 Table of Contents

This subclause is informative.

End of informative text.

22.5.2 Elements

The following information describes the elements in this namespace:

22.5.2.1 datastoreItem (Custom XML Data Properties)

This element specifies the properties for a single custom XML part inside of an Office Open XML document. The set of properties specified within this element are attached to the custom XML part that specifies a relationship to this part.

[*Example*: Consider the following content for a custom XML part properties part:

```
<ds:datastoreItem ds:itemID="{A67AC88A-A164-4ADE-8889-8826CE44DE6E}">
  <ds:schemaRefs>
    <ds:schemaRef ds:uri="http://www.example.com/exampleSchema" />
  </ds:schemaRefs>
</ds:datastoreItem>
```

The datastoreItem element contains the properties for the custom XML part that referenced it; specifically, a part ID of `A67AC88A-A164-4ADE-8889-8826CE44DE6E`, and a single XML Schema reference to a schema with a target namespace of `http://www.example.com/exampleSchema`. *end example*]

Attributes	Description

itemID (Custom XML Data ID)	Specifies a globally unique identifier (GUID) that uniquely identifies a single custom XML part within an Office Open XML document.
	
	Each itemID value shall be unique among all custom XML data parts in this document. If a document contains duplicate itemID values, then the first value should be persisted, and subsequent values should be reassigned.
	
	[*Example*: Consider the following content for a custom XML part properties part:
	
	`<w:datastoreItem w:itemID="{A67AC88A-A164-4ADE-8889-8826CE44DE6E}">`
	`…`
	`</w:datastoreItem>`

Attributes	Description
	The itemID attribute specifies that the ID associated with the parent custom XML part is A67AC88A-A164-4ADE-8889-8826CE44DE6E. *end example*] The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DatastoreItem) is located in §A.6.5. *end note*]

22.5.2.2 schemaRef (Associated XML Schema)

This element specifies a single XML schema that is associated with the custom XML data part. This XML schema is identified using its target namespace, and can be located via any means available to an application processing the contents of this file.

If the custom XML part cannot be validated using the specified XML schema when it is opened, then this reference can be omitted when the file is subsequently saved.

[*Example*: Consider the following content for a custom XML part properties part:

```
<ds:datastoreItem ds:itemID="{A67AC88A-A164-4ADE-8889-8826CE44DE6E}">
  <ds:schemaRefs>
    <ds:schemaRef ds:uri="http://www.example.com/exampleSchema" />
  </ds:schemaRefs>
</ds:datastoreItem>
```

The schemaRef element contains a single XML Schema reference to a schema with a target namespace of http://www.example.com/exampleSchema. Applications can then locate and utilize a schema for this namespace using any means available. *end example*]

Attributes	Description
uri (Target Namespace of Associated XML Schema)	Specifies the target namespace for the XML Schema associated with this schema reference. [*Example*: Consider the following content for a custom XML part properties part: … <w:schemaRef w:uri="http://www.example.com/schema1" /> <w:schemaRef w:uri="http://www.example.com/schema2" /> … The uri attribute specifies the target namespace of each XML schema reference: • http://www.example.com/schema1 • http://www.example.com/schema2

Attributes	Description
	Applications can then locate and utilize a schema for these namespaces using any means available. *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_DatastoreSchemaRef) is located in §A.6.5. *end note*]

22.5.2.3 schemaRefs (Set of Associated XML Schemas)

This element specifies the set of XML schemas that are associated with the parent custom XML part. Any number of XML schemas can be referenced, and this collection of schemas shall then be used to validate the contents of the corresponding custom XML part. If this element is present, then the set of XML schemas provided within should be used to validate the contents of the corresponding custom XML part (including the explicit presence of no child elements to specify that no custom XML schemas should be used even if one is present).

If this element is omitted, then applications can determine the set of XML schemas to be used to validate the contents of this part using any desired means.

[*Example*: Consider the following content for a custom XML part properties part:

```
<ds:datastoreItem ds:itemID="{A67AC88A-A164-4ADE-8889-8826CE44DE6E}">
  <ds:schemaRefs>
    <ds:schemaRef ds:uri="http://www.example.com/exampleSchema" />
  </ds:schemaRefs>
</ds:datastoreItem>
```

The schemaRefs element contains the set of XML Schema references that can be used to validate the contents of this part. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DatastoreSchemaRefs) is located in §A.6.5. *end note*]

22.6 Bibliography

Within an Office Open XML document, it is possible to store an arbitrary amount of bibliographic data, the use of which can be determined by the application reading the content. This subclause defines the format and structure of that bibliographic data.

The outermost element of bibliographic data is Sources, which represents the collection of individual reference materials (Source) in the document.

22.6.1 Table of Contents

This subclause is informative.

End of informative text.

22.6.2 Elements

The following elements define the contents of the Bibliography schema:

22.6.2.1 AbbreviatedCaseNumber (Abbreviated Case Number)

This element describes the abbreviated form of a case number. Typically, this field is used in the Case source type.

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.2 AlbumTitle (Album Title)

This element specifies the title of an album. Typically, this field is used in the Sound recording source type. [*Example*:

```
<b:Source>
  <b:Tag>Bon96</b:Tag>
  <b:SourceType>SoundRecording</b:SourceType>
  <b:Author>
    <b:Performer>
      <b:NameList>
        <b:Person>
          <b:Last>Villaron</b:Last>
          <b:First>Shawn</b:First>
          <b:Middle>Alan</b:Middle>
        </b:Person>
      </b:NameList>
    </b:Performer>
  </b:Author>
  <b:Title>Title</b:Title>
  <b:Year>2004</b:Year>
  <b:City>London</b:City>
  <b:AlbumTitle>Album Title</b:AlbumTitle>
  <b:RefOrder>15</b:RefOrder>
  <b:Guid>{17722923-790D-47E7-BB5D-C5DC67FA83D6}</b:Guid>
  <b:LCID>0</b:LCID>
  <b:Comments>Comments</b:Comments>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.3 Artist (Artist)

This element specifies the artist of the source. Typically, this field is used in the Art and Sound Recording source types. [*Example*:


```
<b:Author>
  <b:Artist>
    <b:NameList>
      <b:Person>
        <b:Last>Jones</b:Last>
        <b:First>Brian</b:First>
      </b:Person>
    </b:NameList>
  </b:Artist>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.4 Author (Contributors List)

This element specifies the contributors to the source. [*Example*:

```
<b:Author>
  <b:Author>
    <b:NameList>
      <b:Person>
        <b:Last>Rothschiller</b:Last>
        <b:First>Chad</b:First>
      </b:Person>
    </b:NameList>
  </b:Author>
  <b:Editor>
    <b:NameList>
      <b:Person>
        <b:Last>Jaeschke</b:Last>
        <b:First>Rex</b:First>
      </b:Person>
    </b:NameList>
  </b:Editor>
```

```
    <b:Translator>
      <b:NameList>
        <b:Person>
          <b:Last>Davis</b:Last>
          <b:First>Tristan</b:First>
        </b:Person>
      </b:NameList>
    </b:Translator>
  </b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_AuthorType) is located in §A.6.6. *end note*]

22.6.2.5 Author (Author)

This element specifies the author of the source. [*Example*:

```
  <b:Author>
    <b:Author>
      <b:NameList>
        <b:Person>
          <b:Last>Jones</b:Last>
          <b:First>Brian</b:First>
        </b:Person>
      </b:NameList>
    </b:Author>
  </b:Author>
```

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. 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_NameOrCorporateType) is located in §A.6.6. *end note*]

22.6.2.6 BookAuthor (Book Author)

This element specifies the author of a book, when the primary author has authored the book section. For example, if person X writes a chapter in a book by person Y, person X is the Author and person Y is the BookAuthor. [*Example*:

```
<b:Author>
  <b:BookAuthor>
    <b:NameList>
      <b:Person>
        <b:Last>Rothschiller</b:Last>
        <b:First>Chad</b:First>
      </b:Person>
    </b:NameList>
  </b:BookAuthor>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.7 BookTitle (Book Title)

This element specifies the title of a book, when the source is a book section. In this case, the title of the book section is the primary title. For example, if X is the title of a chapter in a book entitled , X is the Title and Y is the BookTitle. [*Example*:

```
<b:Source>
   …
  <b:Year>1992</b:Year>
  <b:City>Paris</b:City>
  <b:Publisher>Publisher</b:Publisher>
  <b:Pages>51-84</b:Pages>
  <b:Comments>Comments</b:Comments>
  <b:BookTitle>Book Title</b:BookTitle>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.8 Broadcaster (Broadcaster)

This element specifies the broadcaster of a source. Typically, this field is used in the Interview source type. [*Example*:

```
<b:Source>
   …
  <b:ProgramTitle>Program Title</b:ProgramTitle>
  <b:Broadcaster>Broadcaster</b:Broadcaster>
  <b:Station>Station</b:Station>
  <b:RefOrder>1</b:RefOrder>
  <b:Title>Title (Interview)</b:Title>
  <b:BroadcastTitle>Broadcast Title</b:BroadcastTitle>
  <b:StateProvince>State or Province</b:StateProvince>
  <b:CountryRegion>Country or Region</b:CountryRegion>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.9 BroadcastTitle (Broadcast Title)

This element specifies the broadcast title of a source. Typically, this field is used in the Interview source type. [*Example*:

```
<b:Source>
   …
  <b:ProgramTitle>Program Title</b:ProgramTitle>
  <b:Broadcaster>Broadcaster</b:Broadcaster>
  <b:Station>Station</b:Station>
  <b:RefOrder>1</b:RefOrder>
  <b:Title>Title (Interview)</b:Title>
  <b:BroadcastTitle>Broadcast Title</b:BroadcastTitle>
  <b:StateProvince>State/Province</b:StateProvince>
  <b:CountryRegion>Country/Region</b:CountryRegion>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.10 CaseNumber (Case Number)

This element specifies the case number of a source. Typically, this field is used in the Case source type. [*Example*:

```
…
<b:Title>Title (Case)</b:Title>
<b:Year>Year</b:Year>
<b:City>Place Published</b:City>
<b:ShortTitle>Short Title</b:ShortTitle>
<b:CaseNumber>Case Number</b:CaseNumber>
<b:Court>Court</b:Court>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.11 ChapterNumber (Chapter Number)

This element specifies the number or index of the chapter being referenced. [*Example*:

```
…
<b:BookTitle>Title</b:BookTitle>
<b:Pages>23-65</b:Pages>
<b:Comments>Comments</b:Comments>
<b:ChapterNumber>6</b:ChapterNumber>
<b:RefOrder>1</b:RefOrder>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.12 City (City)

This element specifies the city in which the source was published, printed, or manufactured. [*Example*:

```
…
<b:Title>Title</b:Title>
<b:Year>1997</b:Year>
<b:City>London</b:City>
<b:Publisher>Publihser</b:Publisher>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.13 Comments (Comments)

This element specifies any additional comments about the source. The documentation style determines whether the comments appear in the bibliography. [*Example*:

```
…
<b:ShortTitle>Short Title</b:ShortTitle>
<b:Comments>Comments</b:Comments>
<b:RefOrder>2</b:RefOrder>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.14 Compiler (Compiler)

This element specifies the person who compiled the information in a source. [*Example*:

```
<b:Author>
  <b:Compiler>
    <b:NameList>
      <b:Person>
        <b:Last>Jones</b:Last>
        <b:First>Brian</b:First>
      </b:Person>
    </b:NameList>
  </b:Compiler>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.15 Composer (Composer)

This element specifies the composer of a sound recording. [*Example*:

```
<b:Author>
  <b:Composer>
    <b:NameList>
      <b:Person>
        <b:Last>Davis</b:Last>
        <b:First>Tristan</b:First>
      </b:Person>
    </b:NameList>
  </b:Composer>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.16 Conductor (Conductor)

This element specifies the conductor of a source. Typically, this field is used in the sound recording source type. [*Example*:

```
<b:Author>
  <b:Conductor>
    <b:NameList>
      <b:Person>
        <b:Last>Jones</b:Last>
        <b:First>Brian</b:First>
      </b:Person>
    </b:NameList>
  </b:Conductor>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.17 ConferenceName (Conference or Proceedings Name)

This element specifies the title of the proceedings from a conference. [*Example*:

```
…
<b:Comments>Comments</b:Comments>
<b:ConferenceName>Conference Name</b:ConferenceName>
<b:RefOrder>9</b:RefOrder>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.18 Corporate (Corporate Author)

This element specifies the corporate author, performer, or any field that can be a name. The element is used when an organization, rather than a person, is used. [*Example*:

```
<b:Author>
  <b:Author>
    <b:Corporate>Corporate Author</b:Corporate>
  </b:Author>
</b:Author>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.19 Counsel (Counsel)

This element specifies the counsel, attorney, or attorneys in a case.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.20 CountryRegion (Country or Region)

This element specifies the country or region of a source. [*Example*:

```
<b:Source>
   …
  <b:ProgramTitle>Program Title</b:ProgramTitle>
  <b:Broadcaster>Broadcaster</b:Broadcaster>
  <b:Station>Station</b:Station>
  <b:RefOrder>1</b:RefOrder>
```



```
    <b:Title>Title (Interview)</b:Title>
    <b:BroadcastTitle>Broadcast Title</b:BroadcastTitle>
    <b:StateProvince>State or Province</b:StateProvince>
    <b:CountryRegion>Country or Region</b:CountryRegion>
  </b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.21 Court (Court)

This element specifies the court in which the case was presented. [*Example*:

```
…
<b:Year>1972</b:Year>
<b:CaseNumber>339 1018</b:CaseNumber>
<b:Court>Supreme Court</b:Court>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.22 Day (Day)

This element specifies the day on which a source was created or published. [*Example*:

```
…
<b:PeriodicalTitle>Periodical Title</b:PeriodicalTitle>
<b:Month>November</b:Month>
<b:Day>10</b:Day>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.23 DayAccessed (Day Accessed)

This element specifies the day of the month a source was accessed. [*Example*:

```
<b:MonthAccessed>October</b:MonthAccessed>
<b:DayAccessed>5</b:DayAccessed>
<b:YearAccessed>2000</b:YearAccessed>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.24 Department (Department)

This element specifies the department in which a source originated, or to which a source was submitted. Typically, this field is used in the Report source type, which includes theses and dissertations. [*Example*:

```
…
<b:Institution>Harvard University</b:Institution>
<b:ThesisType>Doctoral Dissertation</b:ThesisType>
<b:Department>Department of Mathematics</b:Department>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.25 Director (Director)

This element specifies the director of a source. Typically, this field is used in the Film source type. [*Example*:

```
<b:Author>
  <b:Director>
    <b:NameList>
      <b:Person>
        <b:Last>Jones</b:Last>
        <b:First>Brian</b:First>
      </b:Person>
    </b:NameList>
  </b:Director>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.26 Distributor (Distributor)

This element specifies the distributor of a source. Typically, this field is used in the Performance and Film source types. [*Example*:

```
…
<b:Distributor>Distributor</b:Distributor>
<b:Country>United States</b:Country>
<b:RefOrder>19</b:RefOrder>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.27 Edition (Editor)

This element specifies the edition of a source. [*Example*:

```
…
<b:Pages>1-34</b:Pages>
<b:Edition>Edition</b:Edition>
<b:Issue>Issue</b:Issue>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.28 Editor (Editor)

This element specifies the editor of a source. [*Example*:

```
<b:Author>
      <b:Editor>
        <b:NameList>
          <b:Person>
            <b:Last>Jaeschke</b:Last>
            <b:First>Rex</b:First>
          </b:Person>
        </b:NameList>
      </b:Editor>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.29 First (Person's First, or Given, Name)

This element specifies a person's first name. [*Example*:

```
<b:Author>
  <b:Editor>
    <b:NameList>
      <b:Person>
        <b:Last>Jaeschke</b:Last>
        <b:First>Rex</b:First>
      </b:Person>
    </b:NameList>
  </b:Editor>
</b:Author>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.30 Guid (GUID)

This element specifies the GUID of a source. [*Example*:

```
<b:Source>
```



```
    …
   <b:RefOrder>2</b:RefOrder>
   <b:Guid>{EE06CBFE-1989-4533-A274-D81DFA436D79}</b:Guid>
   <b:LCID>0</b:LCID>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.31 Institution (Institution)

This element specifies the institution of the source. Typically, this field is used in the Report source type, where it signifies the university or institute, and in the Art source type, where it signifies the museum or institution where the art is housed. [*Example*:

```
<b:Source>
    …
   <b:Institution>Harvard University</b:Institution>
   <b:ThesisType>Dissertation</b:ThesisType>
   <b:RefOrder>12</b:RefOrder>
   <b:Guid>{6CB80970-81D3-476D-90D5-5C9D64E77FAF}</b:Guid>
   <b:LCID>0</b:LCID>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.32 InternetSiteTitle (Internet Site Title)

This element specifies the title of an internet site. Typically, this field is used in the Internet Site and Document from Internet Site source types. [*Example*:

```
…
<b:Title>Title</b:Title>
<b:InternetSiteTitle>Internet Site Title</b:InternetSiteTitle>
<b:Month>July</b:Month>
<b:Day>1</b:Day>
<b:Year>2001</b:Year>
<b:MonthAccessed>Sept.</b:MonthAccessed>
<b:DayAccessed>22</b:DayAccessed>
<b:YearAccessed>1999</b:YearAccessed>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.33 Interviewee (Interviewee)

This element specifies the person being interviewed. Typically, this field is used in the Interview source type. [*Example*:

```
<b:Interviewee>
  <b:NameList>
    <b:Person>
      <b:Last>Rothschiller</b:Last>
      <b:First>Chad</b:First>
    </b:Person>
  </b:NameList>
</b:Interviewee>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.34 Interviewer (Interviewer)

This element specifies the person conducting an interview. Typically, this field is used in the Interview source type. [*Example*:


```
<b:Interviewer>
  <b:NameList>
    <b:Person>
      <b:Last>Davis</b:Last>
      <b:First>Tristan</b:First>
     </b:Person>
  </b:NameList>
</b:Interviewer>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.35 Inventor (Inventor)

This element specifies the inventor of a source. Typically, this field is used in the Patent source type. [*Example*:

```
<b:Author>
  <b:Inventor>
    <b:NameList>
      <b:Person>
        <b:Last>Jones</b:Last>
        <b:First>Brian</b:First>
      </b:Person>
    </b:NameList>
  </b:Inventor>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.36 Issue (Issue)

This element specifies the issue of a source. Typically, this field is used in the Journal Article and Article in Periodical source types. [*Example*:

```
<b:Source>
   …
  <b:Edition>Edition</b:Edition>
  <b:Issue>Issue</b:Issue>
  <b:RefOrder>28</b:RefOrder>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.37 JournalName (Journal Name)

This element specifies the name of the journal. Typically, this field is used in the Journal Article source type. [*Example*:

```
…
<b:Title>Article Title</b:Title>
<b:Year>2000</b:Year>
<b:ShortTitle>Short Title</b:ShortTitle>
<b:Volume>100</b:Volume>
<b:Comments>Comments</b:Comments>
<b:JournalName>Journal Name</b:JournalName>
<b:Pages>91-160</b:Pages>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.38 Last (Person's Last, or Family, Name)

This element specifies a person's last name. [*Example*:

```
<b:Author>
  <b:Editor>
    <b:NameList>
      <b:Person>
        <b:Last>Jaeschke</b:Last>
        <b:First>Tristan</b:First>
      </b:Person>
    </b:NameList>
  </b:Editor>
</b:Author>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.39 LCID (Locale ID)

This element specifies the locale ID of a source, representing the source's language. The set of locale IDs shall be as specified in §22.9.2.6. [*Example*:

```
<b:LCID>ja-JP</b:LCID>
```

end example]

The possible values for this element are defined by the ST_Lang simple type (§22.9.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Lang) is located in §A.6.9. *end note*]

22.6.2.40 Medium (Medium)

This element specifies the medium on or in which a source was created. Typically, this field is used in the Electronic source, sound recording, and film source types. [*Example*:

```
<b:Source>
  …
  <b:LCID>0</b:LCID>
  <b:Medium>DVD</b:Medium>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.41 Middle (Person's Middle, or Other, Name)

This element specifies a person's middle name. [*Example*:

```
<b:Author>
  <b:Editor>
    <b:NameList>
      <b:Person>
        <b:Last>Villaron</b:Last>
        <b:First>Shawn</b:First>
        <b:Middle>Alan</b:Middle>
      </b:Person>
    </b:NameList>
  </b:Editor>
</b:Author>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.42 Month (Month)

This element specifies the month in which a source was created or published. [*Example*:

```
…
<b:PeriodicalTitle>Time</b:PeriodicalTitle>
<b:Month>November</b:Month>
<b:Day>10</b:Day>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.43 MonthAccessed (Month Accessed)

This element specifies the month during which the source was accessed. [*Example*:

```
…
<b:MonthAccessed>October</b:MonthAccessed>
<b:DayAccessed>5</b:DayAccessed>
<b:YearAccessed>2000</b:YearAccessed>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.44 NameList (Name List)

This element specifies a list containing one or more names of a type of contributor to a source, such as a list of authors, editors, or translators. [*Example*:


```
<b:Author>
  <b:Author>
    <b:NameList>
      <b:Person>
        <b:Last>Davis</b:Last>
        <b:First>Tristan</b:First>
      </b:Person>
    </b:NameList>
  </b:Author>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameListType) is located in §A.6.6. *end note*]

22.6.2.45 NumberVolumes (Number of Volumes)

This element specifies the number of volumes a source contains. [*Example*:

```
<b:Source>
   …
  <b:NumberVolumes>10</b:NumberVolumes>
  <b:Comments>Comments</b:Comments>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.46 Pages (Pages)

This element specifies the page range being cited in a source. [*Example*:

```
…
<b:Title>Article Title</b:Title>
<b:Year>2000</b:Year>
<b:ShortTitle>Short Title</b:ShortTitle>
<b:Volume>100</b:Volume>
<b:Comments>Comments</b:Comments>
<b:JournalName>Journal Name</b:JournalName>
<b:Pages>91-160</b:Pages>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.47 PatentNumber (Patent Number)

This element specifies the patent number of a source. Typically, this field is used in the Patent source type. [*Example*:

```
<b:Source>
  …
  <b:PatentNumber>1,000,000</b:PatentNumber>
  <b:RefOrder>26</b:RefOrder>
  <b:Guid>{8295ABC5-2DFD-4FA7-A2A7-A748917C1755}</b:Guid>
  <b:LCID>0</b:LCID>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.48 Performer (Performer)

This element specifies the performer. Typically, this field is used in the sound recording, performance, and film source types. [*Example*:

```
<b:Performer>
  <b:NameList>
    <b:Person>
      <b:Last>Rothschiller</b:Last>
      <b:First>Chad</b:First>
    </b:Person>
  </b:NameList>
</b:Performer>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameOrCorporateType) is located in §A.6.6. *end note*]

22.6.2.49 PeriodicalTitle (Periodical Title)

This element specifies the title of a periodical. [*Example*:


```
…
<b:PeriodicalTitle>Periodical Title</b:PeriodicalTitle>
<b:Month>July</b:Month>
<b:Day>1</b:Day>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.50 Person (Person)

This element specifies a person who contributed to a source. [*Example*:

```
<b:Author>
  <b:NameList>
    <b:Person>
      <b:Last>Villaron</b:Last>
      <b:First>Shawn</b:First>
      <b:Middle>Alan</b:Middle>
    </b:Person>
  </b:NameList>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_PersonType) is located in §A.6.6. *end note*]

22.6.2.51 ProducerName (Producer Name)

This element specifies the person who produced a source. Typically, this field is used in the Internet site, Doc from internet site, Electronic source, Sound recording, Performance, and Film source types. [*Example*:

```
<b:ProducerName>
  <b:NameList>
    <b:Person>
      <b:Last>Rothschiller</b:Last>
      <b:First>Chad</b:First>
    </b:Person>
  </b:NameList>
</b:ProducerName>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.52 ProductionCompany (Production Company)

This element specifies the company that produced a source. Typically, this field is used in the Internet site, Document from internet site, Electronic source, Sound recording, Performance, and Film source types. [*Example*:

```
<b:Source>
   …
  <b:City>Chicago</b:City>
  <b:ProductionCompany>Production Company</b:ProductionCompany>
  <b:Medium>CD</b:Medium>
  <b:RefOrder>16</b:RefOrder>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.53 PublicationTitle (Publication Title)

This element specifies the title of the publication that contains the source. Typically, this field is used in the Electronic Source source type. [*Example*:

```
…
<b:Volume>Volume</b:Volume>
<b:PublicationTitle>Publication Title</b:PublicationTitle>
<b:Month>June</b:Month>
<b:Day>2</b:Day>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.54 Publisher (Publisher)

This element specifies the publisher of a source.

[*Example*:


```
…
<b:City>London</b:City>
<b:Publisher>Publisher</b:Publisher>
<b:ShortTitle>Short Title</b:ShortTitle>
<b:Volume>Volume</b:Volume>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.55 RecordingNumber (Recording Number)

This element specifies the recording number of a source. Typically, this field is used in the sound recording source type.

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.56 RefOrder (Reference Order)

This element specifies the reference order of a source. [*Example*:

```
<b:Source>
    …
  <b:City>Chicago</b:City>
  <b:ProductionCompany>Production Company</b:ProductionCompany>
  <b:Medium>CD</b:Medium>
  <b:RefOrder>16</b:RefOrder>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.57 Reporter (Reporter)

This element specifies the reporter of a source. Typically, this field is used in the Case source type. [*Example*:

```
<b:Source>
```

```
    …
   <b:Reporter>Reporter</b:Reporter>
   <b:RefOrder>27</b:RefOrder>
   <b:Guid>{CE314AB7-E824-4D10-B295-044C68EBED27}</b:Guid>
   <b:LCID>0</b:LCID>
 </b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.58 ShortTitle (Short Title)

This element specifies the short title of a source. [*Example*:

```
…
<b:City>London</b:City>
<b:Publisher>Publisher</b:Publisher>
<b:ShortTitle>Short Title</b:ShortTitle>
<b:Volume>Volume</b:Volume>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.59 Source (Source)

This element specifies the bibliography entry for a source or reference work. [*Example*:

```
<b:Source>
    …
   <b:NumberVolumes>10</b:NumberVolumes>
   <b:Comments>Comments</b:Comments>
 </b:Source>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SourceType) is located in §A.6.6. *end note*]

22.6.2.60 Sources (Sources)

This element specifies the sources in a collection.

Attributes	Description
SelectedStyle (Selected Style)	Specifies the filename of a file which can be used to format the bibliographies and 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://purl.oclc.org/ooxml/bibliographicStyle/APA">` *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
StyleName (Documentation Style Name)	Specifies the name of the documentation style in which the bibliography and citations are formatted. The following values shall be well-defined:

Value	Reference
APA	American Psychological Association. http://www.apa.org/. Publication Manual of the American Psychological Association, Fifth Edition.
Chicago	Chicago Manual of Style, 15th Edition GB7714: GB7714-1987, Standardization Administration of China, 1987-05-05 (http://www.sac.gov.cn)
GOST - Name Soft	ГОСТ 7.1-2003 (GOST 7.1-2003) - The Federal Agency of the Russian Federation on Technical Regulating and Metrology - published by ИПК Издательство стандартов 2004
GOST - Title Sort	ГОСТ 7.1-2003 (GOST 7.1-2003) - The Federal Agency of the Russian Federation on Technical Regulating and Metrology - published by ИПК Издательство стандартов 2004
ISO 690 - First Element and Date	ISO 690-1987(E)-International Organization for Standardization-Second Edition 1987-08-15 (http://www.iso.org) ISO 690-2:1997(E)-International Organization for Standardization-First Edition 1997-11-15 (http://www.iso.org)
ISO 690 - Numerical Reference	ISO 690-1987(E)-International Organization for Standardization-Second Edition 1987-08-15 (http://www.iso.org) ISO 690-2:1997(E)-International Organization for Standardization-

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>First Edition 1997-11-15 (http://www.iso.org)</td></tr>
<tr><td>MLA</td><td>Modern Language Association. http://www.mla.org/. MLA Handbook for Writers of Research Papers, Sixth Edition.</td></tr>
<tr><td>SIST02</td><td>Standard for Information of Science and Technology by Japan Science and Technology Agency, 2003(http://www.jst.go.jp/SIST/handbook/sist02sup/index.htm.</td></tr>
<tr><td>Turabian</td><td>A Manual for Writers of Term Papers, Theses, and Dissertations (Chicago Guides to Writing, Editing, and Publishing), by Kate L. Turabian, 1996.</td></tr>
<tr><td>Any other value</td><td>Implementation-defined.</td></tr>
</table>
[Example:
<pre><b:Sources SelectedStyle="\APA.XSL" StyleName="APA" URI="123"></pre>
end example]
The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
</td></tr>
<tr><td>URI (Uniform Resource Identifier)</td><td>
Specifies a URI or unique identifier with which a documentation style is associated; can be used to uniquely identify versions of styles that share a StyleName.
[Example:
<pre><b:Sources SelectedStyle="\APA.XSL" StyleName="APA" URI="123"></pre>
end example]
The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
</td></tr>
</table>

[*Note*: The W3C XML Schema definition of this element's content model (CT_Sources) is located in §A.6.6. *end note*]

22.6.2.61 SourceType (Source Type)

This element specifies the type of source being cited.

The possible values for this element are defined by the ST_SourceType simple type (§22.6.3.1).

[*Note*: The W3C XML Schema definition of this element's content model (ST_SourceType) is located in §A.6.6. *end note*]

22.6.2.62 StandardNumber (Standard Number)

This element specifies the standard number, such as ISBN or ISSN, of a source. [*Example*:

```
<b:Source>
   …
  <b:NumberVolumes>10</b:NumberVolumes>
  <b:StandardNumber>ISBN or ISSN</b:StandardNumber>
  <b:Comments>Comments</b:Comments>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.63 StateProvince (State or Province)

This element specifies the state or province in which a source was created or published. [*Example*:

```
<b:Source>
   …
  <b:ProgramTitle>Program Title</b:ProgramTitle>
  <b:Broadcaster>Broadcaster</b:Broadcaster>
  <b:Station>Station</b:Station>
  <b:RefOrder>1</b:RefOrder>
  <b:Title>Title (Interview)</b:Title>
  <b:BroadcastTitle>Broadcast Title</b:BroadcastTitle>
  <b:StateProvince>State/Province</b:StateProvince>
  <b:CountryRegion>Country/Region</b:CountryRegion>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.64 Station (Station)

This element specifies the station on which an interview was broadcasted. Typically, this field is used in the Interview source type. [*Example*:

```
…
<b:Month>November</b:Month>
<b:Day>18</b:Day>
<b:Broadcaster>ABC</b:Broadcaster>
<b:Station>WABC</b:Station>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.65 Tag (Tag)

This element specifies the tag name of a source. [*Example*:

```
<b:Source>
  <b:Tag>New01</b:Tag>
  …
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.66 Theater (Theater)

This element specifies the theater in which a source was performed or viewed. Typically, this field is used in the Performer source type. [*Example*:

```
…
<b:Theater>Theater Name</b:Theater>
<b:Month>October</b:Month>
<b:Day>25</b:Day>
<b:RefOrder>19</b:RefOrder>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.67 ThesisType (Thesis Type)

This element specifies the type of report being cited, such as Thesis, Dissertation, or Book Report. Typically, this field is used in the Report source type. [*Example*:

```
…
<b:Institution>Harvard University</b:Institution>
<b:ThesisType>Doctoral Dissertation</b:ThesisType>
<b:Department>Department of Mathematics</b:Department>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.68 Title (Title)

This element specifies the title of a source. [*Example*:

```
…
</b:Author>
<b:Title>Title</b:Title>
<b:Year>2005</b:Year>
<b:City>Seattle</b:City>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.69 Translator (Translator)

This element specifies the translator of a source. [*Example*:

```
<b:Author>
  <b:Translator>
    <b:NameList>
      <b:Person>
        <b:Last>Davis</b:Last>
        <b:First>Tristan</b:First>
      </b:Person>
    </b:NameList>
  </b:Translator>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.70 Type (Patent Type)

This element specifies the type of patent. Typically, this field is used in the Patent source type.

[*Example*:

```
<Source>
  <b:Type>Patent Type</b:Type>
  <b:Guid>{8295ABC5-2DFD-4FA7-A2A7-A748917C1755}</b:Guid>
  <b:LCID>0</b:LCID>
</Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.71 URL (URL)

This element specifies the URL of the source. Typically, this field is used in the Internet Site and Document from Internet Site source types. [*Example*:

```
…
<b:MonthAccessed>September</b:MonthAccessed>
<b:DayAccessed>1</b:DayAccessed>
<b:YearAccessed>1998</b:YearAccessed>
<b:URL>URL</b:URL>
<b:RefOrder>29</b:RefOrder>
…
```


end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.72 Version (Version)

This element specifies the version of the source. Typically, this field is used in the Internet Site and Document from Internet Site source types. [*Example*:

```
<b:Source>
    …
  <b:Version>3.0</b:Version>
  <b:RefOrder>31</b:RefOrder>
  <b:Guid>{F06D8D48-7FD7-4515-88E9-EC70AB9BE792}</b:Guid>
  <b:LCID>0</b:LCID>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.73 Volume (Volume)

This element specifies the volume of the source. [*Example*:

```
…
<b:Title>Article Title</b:Title>
<b:Year>2000</b:Year>
<b:ShortTitle>Short Title</b:ShortTitle>
<b:Volume>100</b:Volume>
<b:Comments>Comments</b:Comments>
<b:JournalName>Journal Name</b:JournalName>
<b:Pages>91-160</b:Pages>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.74 Writer (Writer)

This element specifies the writer of the source. Typically, this field is used in the Performance and Film source types. [*Example*:

```
<b:Author>
  <b:Writer>
    <b:NameList>
      <b:Person>
        <b:Last>Jones</b:Last>
        <b:First>Brian</b:First>
      </b:Person>
    </b:NameList>
  </b:Writer>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.75 Year (Year)

This element specifies the year in which a source was created or published.

[*Example*:

```
…
<b:Title>Title</b:Title>
<b:InternetSiteTitle>Internet Site Title</b:InternetSiteTitle>
<b:Month>July</b:Month>
<b:Day>1</b:Day>
<b:Year>2001</b:Year>
<b:MonthAccessed>Sept.</b:MonthAccessed>
<b:DayAccessed>22</b:DayAccessed>
<b:YearAccessed>1999</b:YearAccessed>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.76 YearAccessed (Year Accessed)

This element specifies the month during which the source was accessed. [*Example*:


```
…
<b:Title>Title</b:Title>
<b:InternetSiteTitle>Internet Site Title</b:InternetSiteTitle>
<b:Month>July</b:Month>
<b:Day>1</b:Day>
<b:Year>2001</b:Year>
<b:MonthAccessed>Sept.</b:MonthAccessed>
<b:DayAccessed>22</b:DayAccessed>
<b:YearAccessed>1999</b:YearAccessed>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.3 Simple Types

This is the complete list of simple types dedicated to Bibliography.

22.6.3.1 ST_SourceType (Bibliographic Data Source Types)

This simple type specifies the possible types of sources that can be used within bibliographic data in an Office Open XML document.

[*Note:* The Office Open XML formats support a collection of predefined source types based on the categories most commonly used in various citation and bibliography style guidelines . The set of predefined source types can be extended as needed. The recommended approach for extending this set is to use the `Misc` type, and then leverage the methods described in ISO/IEC 29500-3 for extending the format with new attributes or elements. *end note.*]

This simple type's contents are a restriction of the ST_String datatype (§22.9.2.13).

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
Art (Art)	Art
ArticleInAPeriodical (Article in a Periodical)	Article in a Periodical
Book (Book)	Book
BookSection (Book Section)	Book Section
Case (Case)	Case
ConferenceProceedings (Conference Proceedings)	Conference Proceedings
DocumentFromInternetSite (Document from Internet Site)	Document from Internet Site

Enumeration Value	Description
ElectronicSource (Electronic Source)	Electronic Source
Film (Film)	Film
InternetSite (Internet Site)	Internet Site
Interview (Interview)	Interview
JournalArticle (Journal Article)	Journal Article
Misc (Miscellaneous)	Miscellaneous
Patent (Patent)	Patent
Performance (Performance)	Performance
Report (Reporter)	Report
SoundRecording (Sound Recording)	Sound Recording

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SourceType) is located in §A.6.6. *end note*]

22.7 Additional Characteristics

In order to allow producers of Office Open XML to describe specific contextual conditions under which the document was created, additional characteristics can be provided within the Additional Characteristics part using the syntax defined below.

The set of additional characteristics is designed to be an extensible list, and can provide a consumer with more information on how to interpret the file. ISO/IEC 29500 defines one set of characteristics; however, additional grammars can be created and associated with a unique URI via the vocabulary attribute.

22.7.1 Table of Contents

This subclause is informative.

End of informative text.

22.7.2 Elements

The following elements define the contents of the Additional Characteristics schema:

22.7.2.1 additionalCharacteristics (Set of Additional Characteristics)

This element is the root element of the Additional Characteristics part and contains the list of additional characteristics for an Office Open XML document.

[*Example:* The following content in an Additional Characteristics part would specify that the producing spreadsheet application supports from 0 to 10,000 columns, and that column ranges should be interpreted accordingly:

```
<additionalCharacteristics>
  <characteristic name="numColumns" relation="le" val="10000"/>
  <characteristic name="numColumns" relation="ge" val="0"/>
</additionalCharacteristics>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_AdditionalCharacteristics) is located in §A.6.7. *end note*]

22.7.2.2 characteristic (Single Characteristic)

This element specifies a single characteristic. The type of characteristic is defined by the name attribute.

[*Example:* A producer can inform the consumer that the computations used to calculate the stored numbers in the formulas belong to a value space expressed by ranges of the binary mantissa and exponent. A consumer can optionally check those characteristics to determine whether, for example, the values should be recalculated. The XML for this would be:

```
<additionalCharacteristics>
  <characteristic name='precisionMantissa' relation='gt'
     val='-9007199254740992'/>
  <characteristic name='precisionMantissa' relation='lt'
     val='9007199254740992'/>
  <characteristic name='precisionExponent' relation='ge' val='-1075'/>
  <characteristic name='precisionExponent' relation='le' val='970'/>
</additionalCharacteristics>
```

end example]

Attributes	Description
name (Name of Characteristic)	Specifies the name of the characteristic. There are no constraints on the value of the name attribute, but each name shall be associated with a specific vocabulary via the vocabulary attribute. The values defined by ISO/IEC 29500 shall be associated with a null vocabulary value, and are as follows:

Attributes	Description

Name Value	Property Specified
numColumns	Number of Columns supported by the spreadsheet producer.
numRows	Number of Rows supported by the spreadsheet producer.
functionVersion	Version of the function specification used
precisionMantissa	Allowed values of the mantissa of numbers within spreadsheet cells/formulas when expressed in base 2.
precisionExponent	Allowed values of the exponent of numbers within spreadsheet cells/formulas when expressed in base 2.
numWorkbookColors	Number of Workbook colors
numConditionalFormatConditions	Number of condition format conditions on a workbook cell
nummaxSortLevels	Number of level of sorting on a range or table
numAutoFilterItems	Number of items shown in the Auto-filter dropdown
numDisplayCellChars	Number of characters that can display in a cell
numPrintCellChars	Number of characters per cell that Excel can print
numUnqiueCellStyles	Number of unique cell styles in a workbook (combinations of all cell formatting)
numFormulaLengthChars	Length of formulas in characters
numFormulaNestingLevel	Number of levels of formula nesting
numFunctionArguments	Number of arguments to a function
numPivotTableRows	Number of rows in a pivot table
numPivotTableColumns	Number of columns in a pivot table
numUniquePivotFieldItems	Number of unique items in a pivot field
numPivotTableMDXNameChars	Number of characters in a MDX name for a pivot table item
numPivotTableRelationChars	String length for a relationship pivot table
numPivotTableFieldLabelChars	Length of field labels in PivotTable including caption length limitations
numPivotTableFields	Number of fields in a pivot table
numSheetXRefArrayFormulas	The number of array formulas in a worksheet that can refer to another (given) worksheet

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

relation (Relationship of Value to Name)	Specifies how the contents of the value attribute should be interpreted in the context of this characteristic.
	
	[*Example:* The following would specify that the application supports from 0 to 10,000 columns, and that column ranges should be interpreted accordingly:
	
	`<additionalCharacteristics>`
	`  <characteristic name="numColumns" relation="le" val="10000"/>`
	`  <characteristic name="numColumns" relation="ge" val="0"/>`
	`</additionalCharacteristics>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Relation simple type (§22.7.3.1).

val (Characteristic Value)	Specifies the value of the characteristic. The possible values for this attribute are defined by the W3C XML Schema string datatype.
vocabulary (Characteristic Grammar)	Specifies a URI defining the characteristic grammar with which the name attribute value shall be interpreted. If this attribute is omitted, then the default grammar (as defined above) shall be used. The possible values for this attribute are defined by the W3C XML Schema anyURI datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Characteristic) is located in §A.6.7. *end note*]

22.7.3 Simple Types

This is the complete list of simple types dedicated to Additional Characteristics.

22.7.3.1 ST_Relation (Characteristic Relationship Types)

This simple type specifies the possible relationships between a characteristic's name and value attributes.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
eq (Equal To)	Equal to.
ge (Greater Than or Equal to)	Greater than or equal to.
gt (Greater Than)	Greater than.
le (Less Than or Equal To)	Less than or equal to.
lt (Less Than)	Less than.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Relation) is located in §A.6.7. *end note*]

22.8 Office Document Relationships

Within an Office Open XML document, it is necessary to be able to explicitly reference one part within the package from another [*Example*: A PresentationML Slide needs to be able to explicitly reference each picture within it to know where each one is anchored. *end example*]

In order to ensure that all such explicit relationship references are easily identifiable within a document, all such relationships are included within attributes in this namespace. This namespace therefore only serves to define attributes used throughout Office Open XML to create explicit relationships, and a single simple type for such attributes.

22.8.1 Table of Contents

This subclause is informative.

End of informative text.

22.8.2 Simple Types

This is the complete list of simple types dedicated to Office Document Relationships.

22.8.2.1 ST_RelationshipId (Explicit Relationship ID)

This simple type specifies the relationship ID in a part's relationship item which is the target of an explicit relationship from the parent XML element.

The kind of relationship which shall be the target of the relationship specified shall be determined based on the context of the parent XML element.

[*Example*: Consider the following markup in an Office Open XML document:


```
<… r:id="rId5" />
```

The id attribute is of type ST_RelationshipID, and therefore the relationship with ID `rId5` must be the target of an explicit relationship from the source part, based on the context of the parent XML element. *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_RelationshipId) is located in §A.6.8. *end note*]

22.9 Shared Simple Types

The following simple types represent common value formats used throughout Office Open XML, and have been centralized in order to ensure their usage remains consistent.

22.9.1 Table of Contents

This subclause is informative.

End of informative text.

22.9.2 Simple Types

This is the complete list of simple types dedicated to Shared Simple Types.

22.9.2.1 ST_CalendarType (Calendar Types)

This simple type specifies the possible types of calendars which can be used within the context of an Office Open XML document.

[*Example*: Consider the following structured document tag properties:

```
<w:sdtPr>
  <w:date w:fullDate="2006-01-01T06:30:00Z">
    <w:calendar w:val="gregorian"/>
  </w:date>
</w:sdtPr>
```

The calendar element specifies that the calendar type for a calendar which can be displayed in the document must be the Gregorian calendar format (gregorian). *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
gregorian (Gregorian)	Specifies that the Gregorian calendar, as defined in ISO 8601, shall be used. This calendar should be localized into the appropriate language.
gregorianArabic (Gregorian Arabic Calendar)	Specifies that the Gregorian calendar, as defined in ISO 8601, shall be used. The values for this calendar should be presented in Arabic.
gregorianMeFrench (Gregorian Middle East French Calendar)	Specifies that the Gregorian calendar, as defined in ISO 8601, shall be used. The values for this calendar should be presented in Middle East French.
gregorianUs (Gregorian English Calendar)	Specifies that the Gregorian calendar, as defined in ISO 8601, shall be used. The values for this calendar should be presented in English.
gregorianXlitEnglish (Gregorian Transliterated English)	Specifies that the Gregorian calendar, as defined in ISO 8601, shall be used. The values for this calendar should be the representation of the English strings in the corresponding Arabic characters (the Arabic transliteration of the English for the Gregorian calendar).

Enumeration Value	Description
gregorianXlitFrench (Gregorian Transliterated French)	Specifies that the Gregorian calendar, as defined in ISO 8601, shall be used. The values for this calendar should be the representation of the French strings in the corresponding Arabic characters (the Arabic transliteration of the French for the Gregorian calendar).
hebrew (Hebrew)	Specifies that the Hebrew lunar calendar, as described by the Gauss formula for Passover [Har'El, Zvi] and The Complete Restatement of Oral Law (Mishneh Torah), shall be used.
hijri (Hijri)	Specifies that the Hijri lunar calendar, as described by the Kingdom of Saudi Arabia, Ministry of Islamic Affairs, Endowments, Da‘wah and Guidance, shall be used.
japan (Japanese Emperor Era)	Specifies that the Japanese Emperor Era calendar, as described by Japanese Industrial Standard JIS X 0301, shall be used.
korea (Korean Tangun Era)	Specifies that the Korean Tangun Era calendar, as described by Korean Law Enactment No. 4, shall be used.
none (No Calendar Type)	Specifies that no calendar should be used.
saka (Saka Era)	Specifies that the Saka Era calendar, as described by the Calendar Reform Committee of India, as part of the Indian Ephemeris and Nautical Almanac, shall be used.
taiwan (Taiwan)	Specifies that the Taiwanese calendar, as defined by the Chinese National Standard CNS 7648, shall be used.
thai (Thai)	Specifies that the Thai calendar, as defined by the Royal Decree of H.M. King Vajiravudh (Rama VI) in Royal Gazette B. E. 2456 (1913 A.D.) and by the decree of Prime Minister Phibunsongkhram (1941 A.D.) to start the year on the Gregorian January 1 and to map year zero to Gregorian year 543 B.C., shall be used.

[*Note*: The W3C XML Schema definition of this simple type’s content model (ST_CalendarType) is located in §A.6.9. *end note*]

22.9.2.2 ST_ConformanceClass (Document Conformance Class Value)

This simple type specifies the conformance class to which a particular Office Open XML document conforms.

[*Example*: Consider the following SpreadsheetML Workbook part markup:

```
<workbook … conformance="transitional">
  …
</workbook>
```

This SpreadsheetML document has a conformance attribute value of `transitional`, therefore it conforms to the SML Transitional conformance class. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
strict (Office Open XML Strict)	Specifies that the document conforms to Office Open XML Strict.
transitional (Office Open XML Transitional)	Specifies that the document conforms to Office Open XML Transitional.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ConformanceClass) is located in §A.6.9. *end note*]

22.9.2.3 ST_FixedPercentage (Fixed Percentage Value with Sign)

This simple type specifies that its contents will contain a percentage measurement from -100% up to and including 100%, including a trailing percent sign (U+0025).

[*Example*: Consider the following WordprocessingML fragment:

```
<w:tcPr>
  <w:tcW w:type="pct" w:w="33.3%" />
</w:pPr>
```

The value of the w attribute is the width of the associated table cell. *end example*]

This simple type's contents are a restriction of the ST_Percentage datatype (§22.9.2.9).

This simple type also specifies the following restrictions:

- This simple type's contents shall match the following regular expression pattern: -?((100)|([0-9][0-9]?))(\.[0-9][0-9]?)?%.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FixedPercentage) is located in §A.6.9. *end note*]

22.9.2.4 ST_Guid (128-Bit GUID)

This simple type specifies that its values shall be a 128-bit globally unique identifier (GUID) value.

[*Example*: Consider the following WordprocessingML fragment for the properties of a single glossary document entry:

```
<w:docPartPr>
  …
  <w:guid w:val="{00000000-5BD2-4BC8-9F70-7020E1357FB2}" />
  …
</w:docPartPr>
```

The guid element specifies that the unique identifier associated with the parent entry shall be {00000000-5BD2-4BC8-9F70-7020E1357FB2}. This value can be used as needed by an application, for example, to uniquely identify a part regardless of its name. *end example*]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type also specifies the following restrictions:

- This simple type's contents shall match the following regular expression pattern: \{[0-9A-F]{8}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{12}\}.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Guid) is located in §A.6.9. *end note*]

22.9.2.5 ST_HexColorRGB (Hexadecimal Color Value)

This simple type specifies that its contents shall contain a color value in RRGGBB hexadecimal format, specified using six hexadecimal digits. Each of the red, green, and blue color values, from 0-255, is encoded as two hexadecimal digits.

[*Example*: Consider a color defined as follows:

```
Red:     122
Green:   23
Blue:    209
```

The resulting RRGGBB value would be 7A17D1, as each color is transformed into its hexadecimal equivalent. *end example*]

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 6 hexadecimal digit(s).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HexColorRGB) is located in §A.6.9. *end note*]

22.9.2.6 ST_Lang (Language Reference)

This simple type specifies that its contents contains a language identifier as defined by RFC 4646/BCP 47.

The contents of this language are interpreted based on the context of the parent XML element.

[*Example*: Consider a language code defined as follows :

```
<w:lang w:val="en-CA" />
```

This language is therefore specified as English (en) and Canada (CA), resulting in use of the English (Canada) language setting. *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_Lang) is located in §A.6.9. *end note*]

22.9.2.7 ST_OnOff (On/Off Value)

This simple type specifies a set of values for any binary (`true` or `false`) property defined in a WordprocessingML document.

A value of `1` or `true` specifies that the property shall be turned on. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted.

A value of `0` or `false` specifies that the property shall be explicitly turned off.

This simple type's contents are a restriction of the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_OnOff) is located in §A.6.9. *end note*]

22.9.2.8 ST_Panose (Panose-1 Number)

This simple type specifies a number consisting of 20 hexadecimal digits which defines the Panose-1 font classification.

[*Example*: Consider the following information stored for a single font:

```
<w:font w:name="Times New Roman">
  <w:panose1 w:val="02020603050405020304" />
  …
</w:font>
```

The panose1 element specifies its Panose-1 number via its val attribute value of `02020603050405020304`. *end example*]

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 20 hexadecimal digit(s).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Panose) is located in §A.6.9. *end note*]

22.9.2.9 ST_Percentage (Percentage Value with Sign)

This simple type specifies that its contents will contain a percentage measurement, with a trailing percent sign (U+0025).

[*Example*: Consider the following WordprocessingML fragment:

```
<w:tcPr>
  <w:tcW w:type="pct" w:w="33.3%" />
</w:pPr>
```

The value of the w attribute is the width of the associated table cell. *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 shall match the following regular expression pattern: -?[0-9]+(\.[0-9]+)?%.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Percentage) is located in §A.6.9. *end note*]

22.9.2.10 ST_PositiveFixedPercentage (Positive Fixed Percentage Value with Sign)

This simple type specifies that its contents will contain a positive percentage measurement from 0% to 100% inclusive, including a trailing percent sign (U+0025).

[*Example*: Consider the following WordprocessingML fragment:

```
<w:tcPr>
  <w:tcW w:type="pct" w:w="33.3%" />
</w:pPr>
```

The value of the w attribute is the width of the associated table cell. *end example*]

This simple type's contents are a restriction of the ST_Percentage datatype (§22.9.2.9).

This simple type also specifies the following restrictions:

- This simple type's contents shall match the following regular expression pattern: ((100)|([0-9][0-9]?))(\.[0-9][0-9]?)?%.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PositiveFixedPercentage) is located in §A.6.9. *end note*]

22.9.2.11 ST_PositivePercentage (Positive Percentage Value with Sign)

This simple type specifies that its contents will contain a positive percentage measurement, including a trailing percent sign (U+0025).

[*Example*: Consider the following WordprocessingML fragment:

```
<w:tcPr>
  <w:tcW w:type="pct" w:w="33.3%" />
</w:pPr>
```

The value of the w attribute is the width of the associated table cell. *end example*]

This simple type's contents are a restriction of the ST_Percentage datatype (§22.9.2.9).

This simple type also specifies the following restrictions:

- This simple type's contents shall match the following regular expression pattern: [0-9]+(\.[0-9]+)?%.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PositivePercentage) is located in §A.6.9. *end note*]

22.9.2.12 ST_PositiveUniversalMeasure (Positive Universal Measurement)

This simple type specifies that its contents will contain a measurement expressed using one of common measure units. The content of this type is a positive decimal number immediately followed by a unit identifier. Unit identifiers are case sensitive and shall be in lowercase. Conforming applications are not required to preserve units of measure between loading and saving a particular document.

The same set of units of measure as in ST_UniversalMeasure type (§22.9.2.15) shall be supported.

This simple type's contents are a restriction of the ST_UniversalMeasure datatype (§22.9.2.15).

This simple type also specifies the following restrictions:

- This simple type's contents shall match the following regular expression pattern: [0-9]+(\.[0-9]+)?(mm|cm|in|pt|pc|pi).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PositiveUniversalMeasure) is located in §A.6.9. *end note*]

22.9.2.13 ST_String (String)

This simple type specifies that its contents contains a string.

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_String) is located in §A.6.9. *end note*]

22.9.2.14 ST_TwipsMeasure (Measurement in Twentieths of a Point)

This simple type specifies that its contents contain wither:

- A positive whole number, whose contents consist of a measurement in twentieths of a point (equivalent to 1/1440th of an inch), or
- A positive decimal number immediately following by a unit identifier.

The contents of this measurement are interpreted based on the context of the parent XML element.

[*Example*: Consider an attribute value of `720` whose type is ST_TwipsMeasure. This attribute value specifies a size of one-half of an inch or 36 points (`720` twentieths of a point = 36 points = 0.5 inches). *end example*]

[*Example*: Consider an attribute value of `12.7mm` whose type is ST_HpsMeasure. This attribute value specifies a size of 0.0127 meter or one-half of an inch or 36 points. *end example*]

This simple type is a union of the following types:

- The ST_PositiveUniversalMeasure simple type (§22.9.2.12).
- The ST_UnsignedDecimalNumber simple type (§22.9.2.16).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TwipsMeasure) is located in §A.6.9. *end note*]

22.9.2.15 ST_UniversalMeasure (Universal Measurement)

This simple type specifies that its contents will contain measurement expressed using one of common measure units. The content of this type is a decimal number immediately followed by a unit identifier. Unit identifiers are case sensitive and shall be in lowercase. Conforming applications are not required to preserve units of measure between loading and saving a particular document.

The following table lists units of measure which are allowed together with their definition based on existing standard or expressed as a conversion from other unit of measure.

Unit Identifier	Definition
`cm`	As defined in ISO 31.
`mm`	As defined in ISO 31.
`in`	1 in = 2.54 cm (informative)
`pt`	1 pt = 1/72 in (informative)
`pc`	1 pc = 12 pt (informative)
`pi`	1 pi = 12 pt (informative)

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 shall match the following regular expression pattern: -?[0-9]+(\.[0-9]+)?(mm|cm|in|pt|pc|pi).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_UniversalMeasure) is located in §A.6.9. *end note*]

22.9.2.16 ST_UnsignedDecimalNumber (Unsigned Decimal Number Value)

This simple type specifies that its contents contain a positive whole decimal number, whose contents are interpreted based on the context of the parent XML element.

[*Example*: Consider the following WordprocessingML fragment:

```
<w:pPr>
  <w:divId w:val="1512645511" />
</w:pPr>
```

The value of the val attribute is the ID of the associated HTML `div`.

However, consider the following fragment:

```
<w:ilvl w:val="1">
  …
</w:ilvl>
```

In this case, the decimal number in the val attribute is the ID of the associated numbering level. In each case, the decimal number value is interpreted in the context of the parent element. *end example*]

This simple type's contents are a restriction of the W3C XML Schema unsignedLong datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_UnsignedDecimalNumber) is located in §A.6.9. *end note*]

22.9.2.17 ST_VerticalAlignRun (Vertical Positioning Location)

This simple type specifies possible values for the alignment of the contents of this run in relation to the default appearance of the run's text. This allows the text to be repositioned as subscript or superscript without altering the font size of the run properties.

[*Example*: Consider a run which must be positioning as superscript when displaying its contents. This requirement would be specified using the following WordprocessingML:

```
<w:rPr>
  <w:vertAlign w:val="superscript" />
</w:rPr>
```


The resulting run is positioned as `superscript`, therefore it is rendered in a smaller size above the default baseline location for the contents of the run. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
baseline (Regular Vertical Positioning)	Specifies that the text in the parent run shall be located at the baseline and presented in the same size as surrounding text.
subscript (Subscript)	Specifies that this text should be subscript. This setting shall lower the text in this run below the baseline and change it to a smaller size, if a smaller size is available.
superscript (Superscript)	Specifies that this text should be superscript. This setting shall raise the text in this run above the baseline and change it to a smaller size, if a smaller size is available.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_VerticalAlignRun) is located in §A.6.9. *end note*]

22.9.2.18 ST_XAlign (Horizontal Alignment Location)

This simple type specifies the set of possible relative horizontal positions for the parent floating object. This relative position is specified relative to the horizontal anchor specified by the parent object.

[*Example*: Consider the following WordprocessingML fragment specifying a text frame:

```
<w:p>
  <w:pPr>
    <w:framePr w:w="2419" w:h="2189" w:hRule="atLeast" w:hSpace="187"
w:wrap="around" w:vAnchor="text" w:hAnchor="page" w:xAlign="left" w:y="73" />
  </w:pPr>
  <w:r>
    <w:t>Text Frame Content.</w:t>
  </w:r>
</w:p>
```

This text frame specifies by the presence of the xAlign attribute to align the frame on the `left` side of the anchor object, in this case, the page. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
center (Centered Horizontally)	Specifies that the parent object shall be centered with respect to the anchor settings. [*Example*: Centered on the page horizontally. *end example*]
inside (Inside)	Specifies that the parent object shall be inside of the anchor object. [*Example*: Inside the text margin horizontally. *end example*]
left (Left Aligned Horizontally)	Specifies that the parent object shall be left aligned with respect to the anchor settings. [*Example*: Left aligned on the page horizontally. *end example*]
outside (Outside)	Specifies that the parent object shall be outside of the anchor object. [*Example*: Outside the text margin horizontally. *end example*]
right (Right Aligned Horizontally)	Specifies that the parent object shall be right aligned with respect to the anchor settings. [*Example*: Right aligned on the page horizontally. *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_XAlign) is located in §A.6.9. *end note*]

22.9.2.19 ST_Xstring (Escaped String)

String of characters with support for escaped invalid-XML characters.

For all characters which cannot be represented in XML as defined by the XML 1.0 specification, the characters are escaped using the Unicode numerical character representation escape character format _xHHHH_, where H represents a hexadecimal character in the character's value. [*Example*: The Unicode character 8 is not permitted in an XML 1.0 document, so it must be escaped as _x0008_. *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_Xstring) is located in §A.6.9. *end note*]

22.9.2.20 ST_YAlign (Vertical Alignment Location)

This simple type specifies the set of possible relative vertical positions for the parent floating object. This relative position is specified relative to the vertical anchor specified by the parent object.

[*Example*: Consider the following WordprocessingML fragment specifying a text frame:

```
<w:p>
  <w:pPr>
    <w:framePr w:w="2419" w:h="2189" w:hRule="atLeast" w:hSpace="187"
w:wrap="around" w:vAnchor="margin" w:hAnchor="page" w:x="1643" w:y="73"
w:yAlign="center" />
  </w:pPr>
  <w:r>
    <w:t>Text Frame Content.</w:t>
  </w:r>
</w:p>
```

This text frame specifies by the presence of the yAlign attribute to vertically align the frame in the `center` of the anchor object, in this case, the `margin`. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bottom (Bottom)	Specifies that the parent object shall be vertically aligned to the bottom edge of the anchor object. [*Example*: At the bottom of the current paragraph. *end example*]
center (Centered Vertically)	Specifies that the parent object shall be vertically centered with respect to the anchor object. Shall not be used with the baseJc element. [*Example*: Centered on the page vertically. *end example*]
inline (In line With Text)	Specifies that the parent object shall be vertically aligned in line with the surrounding text (i.e. shall not allow any text wrapping around it when positioned in the document. Shall not be used with the baseJc element.
inside (Inside Anchor Extents)	Specifies that the parent object shall be vertically aligned to the edge of the anchor object, and positioned inside that object. Shall not be used with the baseJc element.

Enumeration Value	Description
	[*Example*: Inside the text margins vertically. *end example*]
outside (Outside Anchor Extents)	Specifies that the parent object shall be vertically aligned to the edge of the anchor object, and positioned outside that object. Shall not be used with the baseJc element. [*Example*: Outside the text margins vertically. *end example*]
top (Top)	Specifies that the parent object shall be vertically aligned to the top edge of the anchor object . [*Example*: At the top of the current paragraph. *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_YAlign) is located in §A.6.9. *end note*]

22.9.2.21 ST_XmlName (XML Name)

This simple type shall contain an XML non-colonized name (NCName).

[*Example*: Consider the following WordprocessingML fragment for the properties of a single custom XML element:

```
<w:customXmlPr>
<w:attr w:name="company"  w:uri="http://schemas.openxmlformats.org/2006/example"
… />
</w: customXmlPr>
```

The attr element specifies that the NCName associated with the attribute name shall be company. *end example*]

This simple type's contents are a restriction of the W3C XML Schema NCName datatype.

23. Custom XML Schema References

This namespace defines the set of properties which define the location and properties associated with one or more custom XML schemas which have been stored within the contents of a Office Open XML document. Collectively, the set of schemas associated with a document's custom XML markup are referred to as that document's *schema library*. The schema library then stores the set of unique XML namespaces used within the document's custom XML markup, and allows applications to 'tag' these namespaces with appropriate metadata.

23.1 Table of Contents

This subclause is informative.

End of informative text.

23.2 Elements

The following information describes the elements in this namespace:

23.2.1 schema (Custom XML Schema Reference)

This element specifies the properties associated with a single XML namespace, for which all known XML schemas shall be loaded in order to validate the custom XML markup stored within this document. These properties can be used appropriately to locate custom XML schema(s) for use with the document. ISO/IEC 29500 does not require any particular XML schema language.

[*Note:* Some examples of XML schema languages that might be used to implement Custom XML Mappings include:

- W3C XML Schema - http://www.w3.org/XML/Schema
- RELAX NG – ISO/IEC 19757-2
- Schematron – ISO/IEC 19757-3
- NVDL – ISO/IEC 19757-4

end note]

[*Example*: Consider a WordprocessingML document which contains custom XML markup in the http://www.example.com namespace. The following content would be displayed in the document's schema library data:

```
<sl:schemaLibrary>
  <sl:schema sl:uri="http://www.example.com" sl:schemaLocation="c:\example.xsd"
/>
</sl:schemaLibrary>
```

The schema element contains the properties for this one XML namespace: in this case, a namespace URI of `http://www.example.com` and a file location of `c:\example.xsd`. *end example*]

[*Note*: This element is not intended to reintroduce transitional schema into the strict conformance class. *end note*]

Attributes	Description

manifestLocation (Supplementary XML File Location)	Specifies the location of a supplementary XML file which can be downloaded and parsed when this document is loaded in order to provide additional application-defined capabilities. The contents of this file are application-defined.
	
	[*Example*: Consider a WordprocessingML document which contains custom XML markup in the http://www.example.com namespace, which is associated with a resource file located at `http://www.example.com/resource.xml`. The following content would be displayed in the document's schema library data:
	
	`<sl:schemaLibrary>`
	`  <sl:schema sl:uri="http://www.example.com"`
	`sl:manifestLocation= "http://www.example.com/resource.xml" />`
	`</sl:schemaLibrary>`
	
	The manifestLocation attribute contains `http://www.example.com/manifest.xml` which is the location of a resource file that can be downloaded for use when this namespace is used. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

schemaLanguage (Schema Language)	Specifies the media type or the root namespace of the schema language.
	
	[*Example*:
	
	`<sl:schema …`
	`schemaLanguage="http:/relaxng.org/ns/structure/1.0" />`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema token datatype.

schemaLocation (Custom XML Schema Location)	Specifies the location of the XML schema file which should be downloaded and parsed when this document is loaded.

Attributes	Description

	[*Example*: Consider a WordprocessingML document which contains custom XML markup in the http://www.example.com namespace, which is defined by an XML schema located at `c:\example.xsd`. The following content would be displayed in the document's schema library data:
	
	`<sl:schemaLibrary>`
	`  <sl:schema sl:uri="http://www.example.com" sl:schemaLocation=`
	`"c:\example.xsd" />`
	`</sl:schemaLibrary>`
	
	The schemaLocation attribute contains `c:\example.xsd` which is the location of the XML schema file used when this namespace is used. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

uri (Custom XML Schema Namespace)	Specifies the target namespace for the XML Schema associated with this schema reference.
	
	[*Example*: Consider the following content for custom XML namespace data:
	
	`…`
	`<sl:schema w:uri="http://www.example.com/schema1" />`
	`<sl:schema w:uri="http://www.example.com/schema2" />`
	`…`
	
	The uri attribute specifies the target namespace of each XML schema reference:
	• `http://www.example.com/schema1`
	• `http://www.example.com/schema2`
	
	Applications can then locate and utilize a schema for these namespaces using any means available. *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_Schema) is located in §A.7. *end note*]

23.2.2 schemaLibrary (Embedded Custom XML Schema Supplementary Data)

This element specifies the set of XML namespaces which have been associated with the contents of the custom XML markup within the current Office Open XML document. Each unique namespace which is referenced within the document can be referenced within this element by a single schema element, regardless of the number of constituent XML schemas which comprise that namespace.

[*Example*: Consider a WordprocessingML document which contains custom XML markup in two distinct namespaces: the http://www.example.com namespace and the http://www.example2.com namespace. If the first namespace is defined by a single XML schema, and the second is defined by five XML schemas (which are cross-referenced using the appropriate XML Schema syntax), the following content would be displayed in the document's schema library XML:

```
<sl:schemaLibrary>
  <sl:schema … />
  <sl:schema … />
</sl:schemaLibrary>
```

The schemaLibrary element contains only two schema elements even though there are six XML schemas in use, as there are only two distinct namespaces for which data is stored. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SchemaLibrary) is located in §A.7. *end note*]

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-Strict.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://purl.oclc.org/ooxml/officeDocument/math"
  xmlns:r="http://purl.oclc.org/ooxml/officeDocument/relationships"
  xmlns:sl="http://purl.oclc.org/ooxml/schemaLibrary/main"
  xmlns:wp="http://purl.oclc.org/ooxml/drawingml/wordprocessingDrawing"
  xmlns="http://purl.oclc.org/ooxml/wordprocessingml/main"
  xmlns:s="http://purl.oclc.org/ooxml/officeDocument/sharedTypes" elementFormDefault="qualified"
  attributeFormDefault="qualified" blockDefault="#all"
  targetNamespace="http://purl.oclc.org/ooxml/wordprocessingml/main">
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/wordprocessingDrawing"
      schemaLocation="dml-wordprocessingDrawing.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/math" schemaLocation="shared-
      math.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/relationships"
      schemaLocation="shared-relationshipReference.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
    <xsd:import namespace=" http://purl.oclc.org/ooxml/schemaLibrary/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="characterSet" type="s:ST_String" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DecimalNumberOrPercent">
        <xsd:union memberTypes="s:ST_Percentage"/>
    </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"/>
    </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: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: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="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:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_JcTable">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="end"/>
            <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: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: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: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_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="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: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: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_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: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="ffData" type="CT_FFData" minOccurs="0" maxOccurs="1"/>
        </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="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" maxOccurs="unbounded"/>
            <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="bottom" type="CT_Border" minOccurs="0"/>
            <xsd:element name="end" 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="bottom" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="end" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
        </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_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="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:complexType name="CT_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="bottom" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="end" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
        </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="bottom" type="CT_Border" minOccurs="0"/>
            <xsd:element name="end" 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: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="spaceForUL" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="balanceSingleByteDoubleByteWidth" 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="adjustLineHeightInTable" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="applyBreakingRules" 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: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="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="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: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: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="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="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_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="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: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_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="name" type="CT_DocPartName" minOccurs="1"/>
            <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://purl.oclc.org/ooxml/spreadsheetml/main"
  xmlns:r="http://purl.oclc.org/ooxml/officeDocument/relationships"
  xmlns:xdr="http://purl.oclc.org/ooxml/drawingml/spreadsheetDrawing"
  xmlns:s="http://purl.oclc.org/ooxml/officeDocument/sharedTypes"
  targetNamespace="http://purl.oclc.org/ooxml/spreadsheetml/main" elementFormDefault="qualified">
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/relationships"
      schemaLocation="shared-relationshipReference.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/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="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: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="required"/>
        <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="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="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: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="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="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="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="d"/>
            <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="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_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="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="sqref" type="ST_Sqref" use="required"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <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="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="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="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_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: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: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="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="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="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://purl.oclc.org/ooxml/presentationml/main"
  xmlns:p="http://purl.oclc.org/ooxml/presentationml/main"
  xmlns:a="http://purl.oclc.org/ooxml/drawingml/main"
  xmlns:r="http://purl.oclc.org/ooxml/officeDocument/relationships"
  xmlns:s="http://purl.oclc.org/ooxml/officeDocument/sharedTypes" elementFormDefault="qualified"
  targetNamespace="http://purl.oclc.org/ooxml/presentationml/main">
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/relationships"
      schemaLocation="shared-relationshipReference.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/main" schemaLocation="dml-main.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/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"/>
```



```
95                <xsd:element name="diamond" type="CT_Empty"/>
96                <xsd:element name="fade" type="CT_OptionalBlackTransition"/>
97                <xsd:element name="newsflash" type="CT_Empty"/>
98                <xsd:element name="plus" type="CT_Empty"/>
99                <xsd:element name="pull" type="CT_EightDirectionTransition"/>
100               <xsd:element name="push" type="CT_SideDirectionTransition"/>
101               <xsd:element name="random" type="CT_Empty"/>
102               <xsd:element name="randomBar" type="CT_OrientationTransition"/>
103               <xsd:element name="split" type="CT_SplitTransition"/>
104               <xsd:element name="strips" type="CT_CornerDirectionTransition"/>
105               <xsd:element name="wedge" type="CT_Empty"/>
106               <xsd:element name="wheel" type="CT_WheelTransition"/>
107               <xsd:element name="wipe" type="CT_SideDirectionTransition"/>
108               <xsd:element name="zoom" type="CT_InOutTransition"/>
109           </xsd:choice>
110           <xsd:element name="sndAc" minOccurs="0" maxOccurs="1" type="CT_TransitionSoundAction"/>
111           <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
112       </xsd:sequence>
113       <xsd:attribute name="spd" type="ST_TransitionSpeed" use="optional" default="fast"/>
114       <xsd:attribute name="advClick" type="xsd:boolean" use="optional" default="true"/>
115       <xsd:attribute name="advTm" type="xsd:unsignedInt" use="optional"/>
116   </xsd:complexType>
117   <xsd:simpleType name="ST_TLTimeIndefinite">
118       <xsd:restriction base="xsd:token">
119           <xsd:enumeration value="indefinite"/>
120       </xsd:restriction>
121   </xsd:simpleType>
122   <xsd:simpleType name="ST_TLTime">
123       <xsd:union memberTypes="xsd:unsignedInt ST_TLTimeIndefinite"/>
124   </xsd:simpleType>
125   <xsd:simpleType name="ST_TLTimeNodeID">
126       <xsd:restriction base="xsd:unsignedInt"/>
127   </xsd:simpleType>
128   <xsd:complexType name="CT_TLIterateIntervalTime">
129       <xsd:attribute name="val" type="ST_TLTime" use="required"/>
130   </xsd:complexType>
131   <xsd:complexType name="CT_TLIterateIntervalPercentage">
132       <xsd:attribute name="val" type="a:ST_PositivePercentage" use="required"/>
133   </xsd:complexType>
134   <xsd:simpleType name="ST_IterateType">
135       <xsd:restriction base="xsd:token">
136           <xsd:enumeration value="el"/>
137           <xsd:enumeration value="wd"/>
138           <xsd:enumeration value="lt"/>
139       </xsd:restriction>
140   </xsd:simpleType>
141   <xsd:complexType name="CT_TLIterateData">
142       <xsd:choice minOccurs="1" maxOccurs="1">
143           <xsd:element name="tmAbs" type="CT_TLIterateIntervalTime"/>
144           <xsd:element name="tmPct" type="CT_TLIterateIntervalPercentage"/>
145       </xsd:choice>
146       <xsd:attribute name="type" type="ST_IterateType" use="optional" default="el"/>
147       <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>
        <xsd:attribute name="concurrent" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="prevAc" type="ST_TLPreviousActionType" use="optional"/>
        <xsd:attribute name="nextAc" type="ST_TLNextActionType" use="optional"/>
```

```
    </xsd:complexType>
    <xsd:complexType name="CT_TLTimeNodeExclusive">
        <xsd:sequence>
            <xsd:element name="cTn" type="CT_TLCommonTimeNodeData" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TLBehaviorAttributeNameList">
        <xsd:sequence>
            <xsd:element name="attrName" type="xsd:string" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLBehaviorAdditiveType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="base"/>
            <xsd:enumeration value="sum"/>
            <xsd:enumeration value="repl"/>
            <xsd:enumeration value="mult"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLBehaviorAccumulateType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="always"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLBehaviorTransformType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="pt"/>
            <xsd:enumeration value="img"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLBehaviorOverrideType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="normal"/>
            <xsd:enumeration value="childStyle"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLCommonBehaviorData">
        <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:element name="attrNameLst" type="CT_TLBehaviorAttributeNameList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="additive" type="ST_TLBehaviorAdditiveType" use="optional"/>
        <xsd:attribute name="accumulate" type="ST_TLBehaviorAccumulateType" use="optional"/>
        <xsd:attribute name="xfrmType" type="ST_TLBehaviorTransformType" use="optional"/>
        <xsd:attribute name="from" type="xsd:string" use="optional"/>
        <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="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="1" 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: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_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="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://purl.oclc.org/ooxml/officeDocument/relationships"
  xmlns:s="http://purl.oclc.org/ooxml/officeDocument/sharedTypes"
  xmlns="http://purl.oclc.org/ooxml/drawingml/main"
  targetNamespace="http://purl.oclc.org/ooxml/drawingml/main" elementFormDefault="qualified">
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/relationships"
      schemaLocation="shared-relationshipReference.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/diagram" schemaLocation="dml-
      diagram.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/chart" schemaLocation="dml-
      chart.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/picture" schemaLocation="dml-
      picture.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/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="s:ST_Percentage"/>
    </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="s:ST_PositivePercentage"/>
    </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="s:ST_FixedPercentage"/>
    </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="s:ST_PositiveFixedPercentage"/>
    </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="s:ST_Percentage"/>
    </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"/>
    </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: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="s:ST_Percentage"/>
    </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="fontAlgn" 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://purl.oclc.org/ooxml/drawingml/picture"
  xmlns:a="http://purl.oclc.org/ooxml/drawingml/main" elementFormDefault="qualified"
  targetNamespace="http://purl.oclc.org/ooxml/drawingml/picture">
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/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://purl.oclc.org/ooxml/drawingml/lockedCanvas"
  xmlns:a="http://purl.oclc.org/ooxml/drawingml/main"
  xmlns:r="http://purl.oclc.org/ooxml/officeDocument/relationships" elementFormDefault="qualified"
  targetNamespace="http://purl.oclc.org/ooxml/drawingml/lockedCanvas">
```

```
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/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://purl.oclc.org/ooxml/drawingml/main"
xmlns:w="http://purl.oclc.org/ooxml/wordprocessingml/main"
xmlns:dpct="http://purl.oclc.org/ooxml/drawingml/picture"
xmlns:r=http://purl.oclc.org/ooxml/officeDocument/relationships
xmlns="http://purl.oclc.org/ooxml/drawingml/wordprocessingDrawing"
targetNamespace="http://purl.oclc.org/ooxml/drawingml/wordprocessingDrawing"
elementFormDefault="qualified">

    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/main" schemaLocation="dml-main.xsd"/>
    <xsd:import schemaLocation="wml.xsd"
      namespace="http://purl.oclc.org/ooxml/wordprocessingml/main"/>
    <xsd:import namespace=http://purl.oclc.org/ooxml/drawingml/picture
      schemaLocation="dml-picture.xsd"/>
    <xsd:import namespace=http://purl.oclc.org/ooxml/officeDocument/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://purl.oclc.org/ooxml/drawingml/main"
  xmlns="http://purl.oclc.org/ooxml/drawingml/spreadsheetDrawing"
  xmlns:r="http://purl.oclc.org/ooxml/officeDocument/relationships"
  targetNamespace="http://purl.oclc.org/ooxml/drawingml/spreadsheetDrawing"
  elementFormDefault="qualified">
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/main" schemaLocation="dml-main.xsd"/>
    <xsd:import schemaLocation="shared-relationshipReference.xsd"
      namespace="http://purl.oclc.org/ooxml/officeDocument/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://purl.oclc.org/ooxml/drawingml/main"
  xmlns:r="http://purl.oclc.org/ooxml/officeDocument/relationships"
  xmlns="http://purl.oclc.org/ooxml/drawingml/chart"
  xmlns:cdr="http://purl.oclc.org/ooxml/drawingml/chartDrawing"
  xmlns:s="http://purl.oclc.org/ooxml/officeDocument/sharedTypes"
  targetNamespace="http://purl.oclc.org/ooxml/drawingml/chart" elementFormDefault="qualified"
  attributeFormDefault="unqualified" blockDefault="#all">
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/relationships"
      schemaLocation="shared-relationshipReference.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/main" schemaLocation="dml-main.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/chartDrawing" schemaLocation="dml-
      chartDrawing.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/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:union memberTypes="ST_HPercentWithSymbol"/>
    </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: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"/>
    </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: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: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"/>
    </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: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"/>
    </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: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"/>
    </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: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"/>
    </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: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"/>
    </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: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"/>
    </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: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: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://purl.oclc.org/ooxml/drawingml/main"
```

```
  xmlns="http://purl.oclc.org/ooxml/drawingml/chartDrawing"
  targetNamespace="http://purl.oclc.org/ooxml/drawingml/chartDrawing" elementFormDefault="qualified">
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/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://purl.oclc.org/ooxml/drawingml/diagram"
  xmlns:a="http://purl.oclc.org/ooxml/drawingml/main"
  xmlns:r="http://purl.oclc.org/ooxml/officeDocument/relationships"
  xmlns:s="http://purl.oclc.org/ooxml/officeDocument/sharedTypes"
  targetNamespace="http://purl.oclc.org/ooxml/drawingml/diagram" elementFormDefault="qualified"
  attributeFormDefault="unqualified">
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/relationships"
      schemaLocation="shared-relationshipReference.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/main" schemaLocation="dml-main.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/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: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 Shared MLs

A.6.1 Math

This schema is available in the file shared-math.xsd.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://purl.oclc.org/ooxml/officeDocument/math"
  xmlns:m="http://purl.oclc.org/ooxml/officeDocument/math"
  xmlns:w="http://purl.oclc.org/ooxml/wordprocessingml/main"
  xmlns:s="http://purl.oclc.org/ooxml/officeDocument/sharedTypes" elementFormDefault="qualified"
  attributeFormDefault="qualified" blockDefault="#all"
  targetNamespace="http://purl.oclc.org/ooxml/officeDocument/math">
```

```
    <xsd:import namespace="http://purl.oclc.org/ooxml/wordprocessingml/main"
      schemaLocation="wml.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/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.6.2 Extended Properties

This schema is available in the file shared-documentPropertiesExtended.xsd.

```
<xsd:schema xmlns="http://purl.oclc.org/ooxml/officeDocument/extendedProperties"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:vt="http://purl.oclc.org/ooxml/officeDocument/docPropsVTypes"
```

```
  targetNamespace="http://purl.oclc.org/ooxml/officeDocument/extendedProperties"
  elementFormDefault="qualified" blockDefault="#all">
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/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.6.3 Custom Properties

This schema is available in the file shared-documentPropertiesCustom.xsd.

```
<xsd:schema xmlns="http://purl.oclc.org/ooxml/officeDocument/customProperties"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:vt="http://purl.oclc.org/ooxml/officeDocument/docPropsVTypes"
  xmlns:s="http://purl.oclc.org/ooxml/officeDocument/sharedTypes"
  targetNamespace="http://purl.oclc.org/ooxml/officeDocument/customProperties" blockDefault="#all"
  elementFormDefault="qualified">
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/docPropsVTypes"
      schemaLocation="shared-documentPropertiesVariantTypes.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/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.6.4 Variant Types

This schema is available in the file shared-documentPropertiesVariantTypes.xsd.

```
<xsd:schema xmlns="http://purl.oclc.org/ooxml/officeDocument/docPropsVTypes"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:s="http://purl.oclc.org/ooxml/officeDocument/sharedTypes"
  targetNamespace="http://purl.oclc.org/ooxml/officeDocument/docPropsVTypes" blockDefault="#all"
  elementFormDefault="qualified">
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/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.6.5 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://purl.oclc.org/ooxml/officeDocument/customXml"
  xmlns:s="http://purl.oclc.org/ooxml/officeDocument/sharedTypes"
  targetNamespace="http://purl.oclc.org/ooxml/officeDocument/customXml" elementFormDefault="qualified"
  attributeFormDefault="qualified" blockDefault="#all">
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/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.6.6 Bibliography

This schema is available in the file shared-bibliography.xsd.

```
<xsd:schema xmlns="http://purl.oclc.org/ooxml/officeDocument/bibliography"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:s="http://purl.oclc.org/ooxml/officeDocument/sharedTypes"
  targetNamespace="http://purl.oclc.org/ooxml/officeDocument/bibliography"
  elementFormDefault="qualified">
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/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"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_PersonType">
        <xsd:sequence>
            <xsd:element name="Last" type="s:ST_String" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="First" type="s:ST_String" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="Middle" type="s:ST_String" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_NameType">
        <xsd:sequence>
            <xsd:element name="NameList" type="CT_NameListType" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_NameOrCorporateType">
        <xsd:sequence>
            <xsd:choice minOccurs="0" maxOccurs="1">
                <xsd:element name="NameList" type="CT_NameListType" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="Corporate" minOccurs="1" maxOccurs="1" type="s:ST_String"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_AuthorType">
        <xsd:sequence>
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="Artist" type="CT_NameType"/>
                <xsd:element name="Author" type="CT_NameOrCorporateType"/>
                <xsd:element name="BookAuthor" type="CT_NameType"/>
                <xsd:element name="Compiler" type="CT_NameType"/>
                <xsd:element name="Composer" type="CT_NameType"/>
                <xsd:element name="Conductor" type="CT_NameType"/>
                <xsd:element name="Counsel" type="CT_NameType"/>
                <xsd:element name="Director" type="CT_NameType"/>
                <xsd:element name="Editor" type="CT_NameType"/>
                <xsd:element name="Interviewee" type="CT_NameType"/>
                <xsd:element name="Interviewer" type="CT_NameType"/>
                <xsd:element name="Inventor" type="CT_NameType"/>
                <xsd:element name="Performer" type="CT_NameOrCorporateType"/>
                <xsd:element name="ProducerName" type="CT_NameType"/>
                <xsd:element name="Translator" type="CT_NameType"/>
                <xsd:element name="Writer" type="CT_NameType"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SourceType">
        <xsd:sequence>
```



```
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="AbbreviatedCaseNumber" type="s:ST_String"/>
                <xsd:element name="AlbumTitle" type="s:ST_String"/>
                <xsd:element name="Author" type="CT_AuthorType"/>
                <xsd:element name="BookTitle" type="s:ST_String"/>
                <xsd:element name="Broadcaster" type="s:ST_String"/>
                <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.6.7 Additional Characteristics

This schema is available in the file shared-additionalCharacteristics.xsd.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://purl.oclc.org/ooxml/officeDocument/characteristics"
  targetNamespace="http://purl.oclc.org/ooxml/officeDocument/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.6.8 Office Document Relationships

This schema is available in the file shared-relationshipReference.xsd.

```
<xsd:schema xmlns="http://purl.oclc.org/ooxml/officeDocument/relationships"
  xmlns:r="http://purl.oclc.org/ooxml/officeDocument/relationships"
```



```
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
  targetNamespace="http://purl.oclc.org/ooxml/officeDocument/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.6.9 Shared Simple Types

This schema is available in the file shared-commonSimpleTypes.xsd.

```
<xsd:schema xmlns="http://purl.oclc.org/ooxml/officeDocument/sharedTypes"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://purl.oclc.org/ooxml/officeDocument/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_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">
        <xs:union memberTypes="xsd:boolean"/>
    </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_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.7 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://purl.oclc.org/ooxml/schemaLibrary/main"
  targetNamespace="http://purl.oclc.org/ooxml/schemaLibrary/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>
```