# 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.

## 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.).

### 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.

#### 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.

##### 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. |
| 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. |
| 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. |
| compatLnSpc | 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. |
| 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. |
| fromWordArt | 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. |
| horzOverflow (Text | 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |

##### 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.

##### normAutofit (Normal AutoFit)

This element specifies that text within the text body should be normally auto-fit to the bounding box. Autofitting 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| fontScale (Font | Specifies the percentage of the original font size to which each run in the text body is 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. |
| lnSpcReduction | 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. |

##### 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.

#### 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.

##### 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.

##### 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.

|  |  |
| --- | --- |
| **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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |

##### 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. | |
| 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. | |
| 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. | |
| bmk (Bookmark | | Specifies the link target name that is used to reference to the proper link properties in a custom XML part within the document. | |
| 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. | |
| 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. | |
| 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. | |
| 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. | |
| kern (Kerning) | Specifies the minimum font size at which character kerning occurs for this text run. | |
| 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. | |
| 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. | |
| 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. | | |
| normalizeH | Specifies the normalization of height that is to be applied to the text run. This is a renderonly 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. | | |
| 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. | | |
| 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*] | | |
| 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. | | |
| strike | Specifies whether a run of text is formatted as strikethrough text. If this attribute is omitted, than no strikethrough is assumed. | | |
| sz (Font Size) | Specifies the size of text within a text run. Whole points are specified in increments of | | |
| u (Underline) | Specifies whether a run of text is formatted as underlined text. If this attribute is omitted, than no underline is assumed. | | |

##### 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.

|  |  |
| --- | --- |
| **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. |
| 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. |

##### 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.

##### 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.

##### 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **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. | | |
| 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. |
| 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. |
| 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. | | |
| hangingPunct (Hanging | 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. | |
| 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 | |
| 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. | | |
| 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. |
| 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. | |
| 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. | |
| 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. | |

##### 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):

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (On/Off Value) | Specifies a boolean value for the property defined by the parent XML element. |

##### 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.

##### 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.

##### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the percentage of the size that the white space should be. |

##### 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.

|  |  |
| --- | --- |
| **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. |

##### 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.

|  |  |
| --- | --- |
| **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. |
| 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. |

##### 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.

#### 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.

##### 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.

|  |  |
| --- | --- |
| **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. |
| panose (Panose Setting) | Specifies the Panose-1 classification number for the current font using the mechanism defined in §5.2.7.17 of ISO/IEC 14496-22. |
| pitchFamily | Specifies the font pitch as well as the font family for the corresponding font. |
| 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 |

##### 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.

|  |  |
| --- | --- |
| **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. |
| 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. |
| bmk (Bookmark | Specifies the link target name that is used to reference to the proper link properties in a custom XML part within the document. | |
| 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. | |
| 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. | |
| 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. | |
| 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. | |
| kern (Kerning) | Specifies the minimum font size at which character kerning occurs for this text run. | |
| 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. | |
| 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. | |
| 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. | |
| normalizeH | Specifies the normalization of height that is to be applied to the text run. This is a renderonly 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. | |
| 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. | |
| 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*] | |
| 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. | |
| strike | Specifies whether a run of text is formatted as strikethrough text. If this attribute is omitted, than no strikethrough is assumed. | |
| sz (Font Size) | Specifies the size of text within a text run. Whole points are specified in increments of | |
| u (Underline) | Specifies whether a run of text is formatted as underlined text. If this attribute is omitted, than no underline is assumed. | |

##### 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.

|  |  |  |
| --- | --- | --- |
| **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. |
| panose (Panose Setting) | Specifies the Panose-1 classification number for the current font using the mechanism defined in §5.2.7.17 of ISO/IEC 14496-22. | |
| pitchFamily | Specifies the font pitch as well as the font family for the corresponding font. | |
| 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. | |

##### highlight (Highlight Color)

This element specifies the highlight color that is present for a run of text.

##### 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.

|  |  |
| --- | --- |
| **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. |
| endSnd (End Sounds) | Specifies if the URL in question should stop all sounds that are playing when it is clicked. |
| highlightClick | 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. |
| history (Add Hyperlink to Page | 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. |
| id (Drawing Object | Specifies the relationship id that when looked up in this slides relationship file contains the target of this hyperlink. This attribute cannot be omitted. |
| 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. |
| 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. |
| 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. |

##### 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.

|  |  |
| --- | --- |
| **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. |
| endSnd (End Sounds) | Specifies if the URL in question should stop all sounds that are playing when it is clicked. |
| highlightClick | 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. |
| history (Add Hyperlink to Page | 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. |
| id (Drawing Object | Specifies the relationship id that when looked up in this slides relationship file contains the target of this hyperlink. This attribute cannot be omitted. |
| 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. |
| 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. |
| 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. |

##### 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.

|  |  |
| --- | --- |
| **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. |
| panose (Panose Setting) | Specifies the Panose-1 classification number for the current font using the mechanism defined in §5.2.7.17 of ISO/IEC 14496-22. |
| pitchFamily | Specifies the font pitch as well as the font family for the corresponding font. |
| 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 |

##### 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.

##### rPr (Text Run Properties)

This element contains all run level text properties for the text runs within a containing paragraph.

|  |  |  |
| --- | --- | --- |
| **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. |
| 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. |
| 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. |
| bmk (Bookmark | | Specifies the link target name that is used to reference to the proper link properties in a custom XML part within the document. |
| 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. |
| 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. | |
| 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. | |
| 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. | |
| kern (Kerning) | | Specifies the minimum font size at which character kerning occurs for this text run. | |
| 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. | |
| 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. |
| 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. |
| normalizeH | | Specifies the normalization of height that is to be applied to the text run. This is a renderonly 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. |
| 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. |
| 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*] |
| 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. | |
| strike | Specifies whether a run of text is formatted as strikethrough text. If this attribute is omitted, than no strikethrough is assumed. | |
| sz (Font Size) | Specifies the size of text within a text run. Whole points are specified in increments of | |
| u (Underline) | Specifies whether a run of text is formatted as underlined text. If this attribute is omitted, than no underline is assumed. | |

##### 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.

|  |  |  |
| --- | --- | --- |
| **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. |
| panose (Panose | Specifies the Panose-1 classification number for the current font using the mechanism defined in §5.2.7.17 of ISO/IEC 14496-22. | |
| pitchFamily | Specifies the font pitch as well as the font family for the corresponding font. | |
| 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. | |

##### 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.

##### uFill (Underline Fill)

This element specifies the fill color of an underline for a run of text.

##### 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.

##### uLn (Underline Stroke)

This element specifies the properties for the stroke of the underline that is present within a run of text.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| algn (Stroke Alignment) | Specifies the alignment to be used for the underline stroke. |
| 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. |
| cmpd (Compound | Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. |
| 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. |

##### 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.

#### 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.

##### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| startAt (Start | Specifies the number that starts a given sequence of automatically numbered bullets. |
| type (Bullet | 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. |

##### 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.

##### 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.

|  |  |
| --- | --- |
| **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. |

##### 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.

##### 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.

##### 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.

|  |  |  |
| --- | --- | --- |
| **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. |
| panose (Panose Setting) | Specifies the Panose-1 classification number for the current font using the mechanism defined in §5.2.7.17 of ISO/IEC 14496-22. | |
| pitchFamily | Specifies the font pitch as well as the font family for the corresponding font. | |
| 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. | |

##### 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.

##### 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.

##### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the percentage of the text size that this bullet should be. This attribute should |

##### 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.

|  |  |
| --- | --- |
| **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. |

##### 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.

##### lstStyle (Text List Styles)

This element specifies the list of styles associated with this body of text.

##### 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **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. | | |
| 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. |
| 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. |
| 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. | | |
| hangingPunct (Hanging | 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. | |
| 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 | |
| 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. | | |
| 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. |
| 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. | |
| 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. | |
| 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. | |

##### 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **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. | | |
| 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. |
| 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. |
| 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. | | |
| hangingPunct (Hanging | 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. | |
| 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. | |
| 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. | | |
| 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. |
| 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. | |
| 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. | |
| 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. | |

##### 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.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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. | | |
| 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. | | |
| 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. | | |
| 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. | | |
| hangingPunct (Hanging | | 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. | | |
| 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. | | |
| 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. | | |
| 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. | | |
| 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. | | |
| 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. | | |
| 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. | | |

##### 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **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. | | |
| 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. |
| 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. |
| 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. | | |
| hangingPunct (Hanging | 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. | |
| 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. | |
| 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. | | |
| 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. |
| 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. | |
| 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. | |
| 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. | |

##### 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.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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. | | |
| 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. | | |
| 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. | | |
| 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. | | |
| hangingPunct (Hanging | | 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. | | |
| 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. | | |
| 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. | | |
| 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. | | |
| 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. | | |
| 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. | | |
| 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. | | |

##### 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **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. | | |
| 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. |
| 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. |
| 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. | | |
| hangingPunct (Hanging | 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. | |
| 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. | |
| 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. | | |
| 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. |
| 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. | |
| 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. | |
| 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. | |

##### 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **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. | | |
| 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. |
| 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. |
| 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. | | |
| hangingPunct (Hanging | 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. | |
| 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. | |
| 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. | | |
| 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. |
| 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. | |
| 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. | |
| 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. | |

##### 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **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. | | |
| 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. |
| 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. |
| 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. | | |
| hangingPunct (Hanging | 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. | |
| 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. | |
| 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. | | |
| 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. |
| 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. | |
| 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. | |
| 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. | |

##### 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **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. | | |
| 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. |
| 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. |
| 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. | | |
| hangingPunct (Hanging | 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. | |
| 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. | |
| 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. | | |
| 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. |
| 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. | |
| 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. | |
| 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. | |

#### 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*.

### Tables

This section contains information regarding the definition of a table within DrawingML. The following image is an example table within DrawingML.

#### cell3D (Cell 3-D)

This element specifies a set of properties which dictate the 3-D appearance of a given cell in a 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. |

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| w (Width) | The width of the column in EMUs. |

#### 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.

#### 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.

#### lnB (Bottom Border Line Properties)

This element defines the line properties associated with the bottom border of a given cell.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| algn (Stroke Alignment) | Specifies the alignment to be used for the underline stroke. |
| 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. |
| cmpd (Compound | Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. |
| 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. |

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| algn (Stroke | Specifies the alignment to be used for the underline stroke. |
| 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. |
| cmpd (Compound | Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. |
| 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. |

#### lnL (Left Border Line Properties)

This element defines the line properties associated with the left border of a cell

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| algn (Stroke Alignment) | Specifies the alignment to be used for the underline stroke. |
| 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. |
| cmpd (Compound | Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. |
| 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. |

#### lnR (Right Border Line Properties)

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

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| algn (Stroke Alignment) | Specifies the alignment to be used for the underline stroke. |
| 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. |
| cmpd (Compound | Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. |
| 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. |

#### lnT (Top Border Line Properties)

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

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| algn (Stroke Alignment) | Specifies the alignment to be used for the underline stroke. |
| 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. |
| cmpd (Compound | Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. |
| 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. |

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| algn (Stroke Alignment) | Specifies the alignment to be used for the underline stroke. |
| 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. |
| cmpd (Compound | Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. |
| 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. |

#### 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:

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| styleId (Style ID) | Specifies a GUID identifying the table style in a unique manner. |
| styleName (Name) | Specifies the name of the table style which can show up in the user interface identifying the style to a user. |

#### 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.

#### 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.

#### 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.

#### 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.

|  |  |  |
| --- | --- | --- |
| **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. | |
| 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. | |
| firstCol (First Column) | Enables or disables the first column formatting for a table style. A value of 1 or true enables the first column formatting defined in the table style. The attribute defaults to false if it is not specified. | |
| 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. |
| 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. |
| 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 |
| 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. |

#### 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.

|  |  |
| --- | --- |
| **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. |
| 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. |
| 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. |
| 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. |
| 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. |

#### 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.

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Description** | |
| anchor (Anchor) | Defines the alignment of the text vertically within the cell. | |
| anchorCtr (Anchor Center) | When this attribute is 1 or true , it modifies the anchor attribute. This attribute centeraligns the text box itself which allows for text to be left aligned along the center of the cell for example. | |
| horzOverflow | 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. | |
| 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. | |
| 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. |
| 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. |
| 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. |
| vert (Text Direction) | Defines the text direction within the cell. |

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| h (Height) | Defines the height of the row in the table. |

## 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.

### 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.

#### applyToEnd (Apply to End)

This element specifies the picture shall be applied to the end of the point or series.

#### 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. |

#### 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. |

#### area3DChart (3D Area Charts)

This element specifies the 3-D area series on this chart.

#### areaChart (Area Charts)

This element specifies the 2-D area series on this chart.

#### 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. |

#### 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. |

#### 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. |

#### 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. |

#### 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. |

#### backWall (Back Wall)

This element specifies the back wall of the chart.

#### 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 nonnegative value.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Floating Point Value) | Specifies that the contents of this attribute contain a floating point number. |

#### bandFmt (Band Format)

This element specifies the formatting band of a surface chart.

#### bandFmts (Band Formats)

This element contains a collection of formatting bands for a surface chart indexed from low to high.

#### bar3DChart (3D Bar Charts)

This element contains the 3-D bar or column series on this chart.

#### barChart (Bar Charts)

This element contains the 2-D bar or column series on this chart.

#### 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. |

#### 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. |

#### 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. |

#### bubbleChart (Bubble Charts)

This element contains the bubble series on this chart.

#### 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. |

#### bubbleSize (Bubble Size)

This element specifies the data for the sizes of the bubbles on the bubble chart.

#### 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. |

#### cat (Category Axis Data)

This element specifies the data used for the category axis.

#### catAx (Category Axis Data)

This element specifies the category axis of the chart.

#### chart (Reference to Chart Part)

This element specifies the chart.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| id (Relationship | 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. |

#### chart (Chart)

This element specifies the chart.

#### 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. |

#### chartSpace (Chart Space)

This element specifies overall settings for a single chart, and is the root node for the chart part.

#### 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) | Specifies a color defined which is associated as the accent 1 color. |
| accent2 (Accent 2) | Specifies a color defined which is associated as the accent 2 color. |
| accent3 (Accent 3) | Specifies a color defined which is associated as the accent 3 color. |
| accent4 (Accent 4) | Specifies a color defined which is associated as the accent 4 color. |
| accent5 (Accent 5) | Specifies a color defined which is associated as the accent 5 color. |
| accent6 (Accent 6) | Specifies a color defined which is associated as the accent 6 color. |
| bg1 (Background 1) | A color defined which is associated as the first background color. |
| bg2 (Background 2) | Specifies a color defined which is associated as the second background color. |
| folHlink (Followed | Specifies a color defined which is associated as the color for a followed hyperlink. |
| hlink (Hyperlink) | Specifies a color defined which is associated as the color for a hyperlink. |
| tx1 (Text 1) | Specifies a color defined which is associated as the first text color. |
| tx2 (Text 2) | Specifies a color defined which is associated as the second text color. |

#### 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. |

#### crossBetween (Cross Between)

This element specifies whether the value axis crosses the category axis between categories.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Cross Between Value) | Specifies whether the value axis crosses the category axis between categories or on categories. |

#### 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. |

#### crossesAt (Crossing Value)

This element specifies where on the axis the perpendicular axis crosses. The units are dependent on the type of axis.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Floating Point Value) | Specifies that the contents of this attribute contain a floating point number. |

#### custSplit (Custom Split)

This element contains the custom split information for a pie-of-pie or bar-of-pie chart with a custom split.

#### 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. |

#### 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. |

#### 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. |

#### dateAx (Date Axis)

This element specifies a date axis for the chart.

#### delete (Delete)

This element specifies that the chart element specified by its containing element shall be deleted from the chart.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Boolean Value) | Specifies a boolean value for the property defined by the parent XML element. |

#### 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. |

#### 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. |

#### dispEq (Display Equation)

This element specifies that the equation for the trendline is displayed on the chart (in the same label as the Rsquared value).

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Boolean Value) | Specifies a boolean value for the property defined by the parent XML element. |

#### 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. |

#### dispUnits (Display Units)

This element specifies the scaling value of the display units for the value axis.

#### dispUnitsLbl (Display Units Label)

This element specifies the display unit label for the value axis in the specified chart.

#### dLbl (Data Label)

This element specifies a data label.

#### 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. |

#### 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.

#### doughnutChart (Doughnut Charts)

This element contains the doughnut chart series.

#### downBars (Down Bars)

This element specifies the down bars.

#### dPt (Data Point)

This element specifies a single data point.

#### dropLines (Drop Lines)

This element specifies drop lines.

#### dTable (Data Table)

This element specifies a data table.

#### errBars (Error Bars)

This element specifies error bars. The errValType element controls whether the minus, plus, or val elements are used.

#### 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. |

#### 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. |

#### 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. |

#### evenFooter (Even Footer)

This element specifies the footer to use on even numbered pages. (See §18.3.1.38 for more information.) The possible values for this element are defined by the ST\_Xstring simple type (§22.9.2.19).

|  |  |
| --- | --- |
| Attributes | Description |
| xml:space (Content | Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. |

#### evenHeader (Even Header)

This element specifies the header to use on even numbered pages. (See §18.3.1.39 for more information.)

|  |  |
| --- | --- |
| Attributes | Description |
| xml:space (Content | Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. |

#### 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. |

#### 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. |

#### externalData (External Data Relationship)

This element specifies the relationship to the data for this chart.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| id (Relationship | Specifies the relationship ID for the relationship for this chart. The relationship explicitly targeted by this attribute shall either be of type |

#### extLst (Chart Extensibility)

This element contains tags used for future extensibility of the file format.

#### 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.

#### firstFooter (First Footer)

This element specifies the footer to use on the first page. (See §18.3.1.41 for more information.) The possible values for this element are defined by the ST\_Xstring simple type (§22.9.2.19).

|  |  |
| --- | --- |
| Attributes | Description |
| xml:space (Content | Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. |

#### firstHeader (First Header)

This element specifies the header to use on the first page. (See §18.3.1.42 for more information.)

|  |  |
| --- | --- |
| Attributes | Description |
| xml:space (Content | Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. |

#### 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. |

#### floor (Floor)

This element specifies the floor of a 3D chart.

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Integer Value) | Specifies that the contents of this attribute contain an integer number. |

#### 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.

#### 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. |

#### 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. |

#### 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%. |

#### 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%. |

#### grouping (Grouping)

This element specifies the kind of grouping for a column, line, or area chart.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Grouping | Specifies the grouping value. |

#### 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. |

#### 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. |

#### headerFooter (Header and Footer)

This element specifies the headers and footers that shall be used when the chart is printed. (See §18.3.1.46 for more information.)

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| alignWithMargins | Specifies the header and footer should align with the left and right margins of the chart. |
| differentFirst (Different First) | Specifies the header and footer are different for the first page. |
| differentOddEven | Specifies the header and footer are different on odd-numbered pages and evennumbered pages. |

#### hiLowLines (High Low Lines)

This element specifies the high-low lines for the series.

#### 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. |

#### 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. |

#### 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%. |

#### 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. |

#### 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. |

#### 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. |

#### 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. |

#### layout (Layout)

This element specifies how the chart element is placed on the chart.

#### 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. |

#### 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. |

#### 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%. |

#### leaderLines (Leader Lines)

This element specifies the leader lines for data labels.

#### legend (Legend)

This element specifies the legend.

#### legendEntry (Legend Entry)

This element specifies a legend entry.

#### legendPos (Legend Position)

This element specifies the position of the legend.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Legend Position Value) | Specifies the position of the legend. |

#### line3DChart (3D Line Charts)

This element contains the 3-D line chart series.

#### lineChart (Line Charts)

This element contains the 2-D line chart series.

#### 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. |

#### lvl (Level)

This element specifies data for a single level of labels for a category axis.

#### majorGridlines (Major Gridlines)

This element specifies major gridlines.

#### majorTickMark (Major Tick Mark)

This element specifies the major tick marks.

|  |  |
| --- | --- |
| Attributes | Description |
| val (Tick Mark Value) | Specifies the major tick mark position. |

#### 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. |

#### 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. |

#### manualLayout (Manual Layout)

This element specifies the exact position of a chart element.

#### 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. |

#### marker (Marker)

This element specifies a data marker.

#### 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. |

#### 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. |

#### minorGridlines (Minor Gridlines)

This element specifies the minor gridlines.

#### 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. |

#### 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. |

#### 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. |

#### minus (Minus)

This element specifies the error bar value in the negative direction. It shall be used only when the errValType is cust.

#### multiLvlStrCache (Multi Level String Cache)

This element specifies the last data shown on the chart for a category axis.

#### multiLvlStrRef (Multi Level String Reference)

This element specifies a reference to data for the category axis with a cache of the last values used.

#### name (Trendline Name)

This element specifies the name of the trendline.

#### name (Pivot Name)

This element specifies the name of the pivot table to get the data for the chart from.

#### 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. |

#### 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. |

#### numCache (Number Cache)

This element specifies the last data shown on the chart for a series.

#### numFmt (Number Format)

This element specifies number formatting for the parent element.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| formatCode (Number Format | 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. |
| sourceLinked | Specifies a boolean value for the property defined by the parent XML element. |

#### numLit (Number Literal)

This element specifies a set of numbers used for the parent element.

#### numRef (Number Reference)

This element specifies a reference to numeric data with a cache of the last values used.

#### oddFooter (Odd Footer)

This element specifies the footer to use on odd numbered pages. (See §18.3.1.57 for more information.)

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| xml:space (Content | Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. |

#### oddHeader (Odd Header)

This element specifies the header to use on odd numbered pages. (See §18.3.1.58 for more information.) The possible values for this element are defined by the ST\_Xstring simple type (§22.9.2.19).

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| xml:space (Content | Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. |

#### 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.

#### 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. |

#### 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. |

#### 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. |

#### 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. |

#### 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%. |

#### 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. |

#### pageMargins (Page Margins)

This element specifies the page margins for a chart.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| b (Bottom) | Specifies the bottom page margin in inches. |
| footer (Footer) | Specifies the footer margin in inches. |
| header (Header) | Specifies the header margin in inches. |
| l (Left) | Specifies the left page margin in inches. |
| r (Right) | Specifies the right page margin in inches. |
| t (Top) | Specifies the top page margin in inches. |

#### pageSetup (Page Setup)

This element defines the page setup for the chart.

|  |  |  |
| --- | --- | --- |
| Attributes | | Description |
| blackAndWhite | | Specifies the page shall print in black and white. |
| copies (Copies) | | Specifies the number of copies that shall be printed. |
| draft (Draft) | | Specifies the page shall be printed in draft mode. |
| firstPageNumber | | Specifies the page number. |
| horizontalDpi | | Specifies the horizontal resolution to print in dots per inch. |
| orientation (Orientation) | | Specifies the orientation of the paper. |
| paperHeight (Paper Height) | | Height of custom paper as a number followed by a unit identifier. [*Example*: 297mm, 11in *end example*] |
| paperSize (Paper Size) | | Specifies the paper size according to the following table. |
| paperWidth (Paper Width) | Width of custom paper as a number followed by a unit identifier. [*Example*: 21cm, 8.5in *end example*] | |
| useFirstPageNumb | Specifies to use the first page number instead of automatically generating a page number. | |
| verticalDpi | Specifies the vertical resolution to print in dots per inch. | |

#### 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. |

#### 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 |
| Value) | and 240, whose unit is one-half degrees. |

#### 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. |

#### pictureOptions (Picture Options)

This element specifies the picture to be used on the data point, series, wall, or floor.

#### 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. |

#### pie3DChart (3D Pie Charts)

This element contains the 3-D pie series for this chart.

#### pieChart (Pie Charts)

This element contains the 2-D pie series for this chart.

#### pivotFmt (Pivot Format)

This element contains a set of formatting to be applied to the chart that is based on a pivotTable.

#### pivotFmts (Pivot Formats)

This element contains a collection of formatting bands for a surface chart indexed from low to high.

#### pivotSource (Pivot Source)

This element specifies the source pivot table for a pivot chart.

#### plotArea (Plot Area)

This element specifies the plot area of the chart.

#### 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. |

#### plus (Plus)

This element specifies the error bar value in the positive direction. It shall be used only when the errValType is cust.

#### printSettings (Print Settings)

This element specifies the print settings for the chart.

#### 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.

#### 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. |
| idx (Index) | The index of the series in the collection |

#### 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. |

#### 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. |

#### radarChart (Radar Charts)

This element contains the radar chart series on this chart.

#### 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. |

#### rAngAx (Right Angle Axes)

This element specifies that the chart axes are at right angles, rather than drawn in perspective. Applies only to 3D charts.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Boolean Value) | Specifies a boolean value for the property defined by the parent XML element. |

#### rich (Rich Text)

This element contains a string with rich text formatting.

#### 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. |

#### 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. |

#### 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. |

#### scaling (Scaling)

This element contains additional axis settings.

#### scatterChart (Scatter Charts)

This element contains the scatter chart series for this chart.

#### 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. |

#### 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. |

#### 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%. |

#### 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. |

#### 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.

#### ser (Scatter Chart Series)

This element specifies a series on a scatter chart.

#### ser (Area Chart Series)

This element specifies a series on an area chart.

#### ser (Radar Chart Series)

This element specifies a series on a radar chart.

#### ser (Bar Chart Series)

This element specifies a series on a bar chart.

#### ser (Line Chart Series)

This element specifies a series on a line chart.

#### ser (Pie Chart Series)

This element specifies a series on a doughnut or pie chart.

#### ser (Surface Chart Series)

This element specifies a series on a surface chart.

#### ser (Bubble Chart Series)

This element specifies a series on a bubble chart.

#### serAx (Series Axis)

This element specifies a series axis for the chart.

#### serLines (Series Lines)

This element specifies series lines for the chart.

#### 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. |

#### 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. |

#### 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. |

#### 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. |

#### 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. |

#### 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. |

#### 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. |

#### 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. |

#### 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. |

#### 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. |

#### 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. |

#### 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. |

#### 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. |

#### 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. |

#### sideWall (Side Wall)

This element specifies the side wall.

#### 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. |

#### 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. |

#### 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. |

#### 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. |

#### 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. |

#### 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 | 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. |

#### stockChart (Stock Charts)

This element contains the collection of stock chart series.

#### strCache (String Cache)

This element specifies the last string data used for a chart.

#### strLit (String Literal)

This element specifies a set of strings used for a chart

#### 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.

#### style (Style)

This element specifies the style that shall be applied to the chart.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Style Type) | Specifies the chart style. |

#### surface3DChart (3D Surface Charts)

This element contains the set of 3-D surface series.

#### surfaceChart (Surface Charts)

This element contains the set of 2-D contour charts.

#### symbol (Symbol)

This element specifies the marker that is used for the data points.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Marker Style Value) | Specifies the marker style. |

#### 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. |

#### 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. |

#### 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. |

#### 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. |

#### title (Title)

This element specifies a title.

#### trendline (Trendlines)

This element specifies a trendline.

#### trendlineLbl (Trendline Label)

This element specifies the label for the trendline.

#### trendlineType (Trendline Type)

This element specifies the style of the trendline.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Trendline Type Value) | Specifies the trendline style. |

#### tx (Chart Text)

This element specifies text to use on a chart, including rich text formatting.

#### tx (Series Text)

This element specifies text for a series name, without rich text formatting.

#### txPr (Text Properties)

This element specifies text formatting. The lstStyle element is not supported.

#### upBars (Up Bars)

This element specifies the up bars on the chart.

#### upDownBars (Up/Down Bars)

This element specifies the up and down bars.

#### 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. |

#### userShapes (User Shapes)

This element shall specify the shapes drawn on top of the chart.

#### 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 | 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. |

#### v (Numeric Value)

This element specifies a numeric value.

#### v (Text Value)

This element specifies a text value for a category axis label or a series name.

#### val (Values)

This element specifies the data values which shall be used to define the location of data markers on a chart.

#### 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. |

#### valAx (Value Axis)

This element specifies a value axis.

#### 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. |

#### view3D (View In 3D)

This element specifies the 3-D view of the chart.

#### 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. |

#### 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. |

#### 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. |

#### 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. |

#### 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. |

#### xVal (X Values)

This element specifies the x values which shall be used to define the location of data markers on a chart.

#### 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. |

#### 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. |

#### yVal (Y Values)

This element specifies the y values which shall be used to define the location of data markers on a chart.

### Simple Types

This is the complete list of simple types dedicated to DrawingML – Charts.

#### ST\_AxisUnit (Axis Unit)

This simple type specifies that its contents contain a positive floating point number.

#### ST\_AxPos (Axis Position)

This simple type specifies the possible positions for an axis.

|  |  |
| --- | --- |
| **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. |
| t (Top) | Specifies that the axis shall be displayed at the top of the plot area. |

#### ST\_BarDir (Bar Direction)

This simple type specifies the possible directions for a bar chart.

|  |  |
| --- | --- |
| **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. |

#### ST\_BarGrouping (Bar Grouping)

This simple type specifies the possible groupings for a bar chart.

|  |  |
| --- | --- |
| **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. |

#### ST\_BubbleScale (Bubble Scale)

This simple type specifies that its contents contain a percentage between 0% and 300%.

#### ST\_BuiltInUnit (Built-In Unit)

This simple type specifies the built in display units for an axis.

|  |  |
| --- | --- |
| **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 1,000,000,000,000. |

#### ST\_CrossBetween (Cross Between)

This simple type specifies the possible crossing states of an axis.

|  |  |
| --- | --- |
| **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. |

#### ST\_Crosses (Crosses)

This simple type specifies the possible crossing points for an axis.

|  |  |
| --- | --- |
| **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. |

#### ST\_DepthPercent (Depth Percent)

This simple type specifies that its contents contain a percentage between 20% and 2000%.

#### ST\_DispBlanksAs (Display Blanks As)

This simple type specifies the possible ways to display blanks.

|  |  |
| --- | --- |
| **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. |

#### ST\_DLblPos (Data Label Position)

This simple type specifies the possible positions for a data label.

|  |  |
| --- | --- |
| **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. |

#### ST\_ErrBarType (Error Bar Type)

This simple type specifies the possible ways to draw an error bar.

|  |  |
| --- | --- |
| **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. |

#### ST\_ErrDir (Error Bar Direction)

This simple type specifies the possible directions for error bars.

|  |  |
| --- | --- |
| **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. |

#### 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.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| cust (Custom Error Bars) | Specifies that the length of the error bars shall be determined by the Plus and Minus elements. |
| 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. |

#### ST\_FirstSliceAng (First Slice Angle)

This simple type specifies that its contents contain an integer between 0 and 360.

#### ST\_GapAmount (Gap Amount)

This simple type specifies that its contents contain a percentage between 0% and 500%.

#### ST\_Grouping (Grouping)

This simple type specifies the possible groupings for a bar chart.

|  |  |
| --- | --- |
| **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. |

#### ST\_HoleSize (Hole Size)

This simple type specifies that its contents contain a percentage between 1% and 90%.

#### ST\_HPercent (Height Percent)

This simple type specifies that its contents contain a percentage between 5% and 500%.

#### ST\_LayoutMode (Layout Mode)

This simple type specifies the possible ways to store a chart element's position.

|  |  |
| --- | --- |
| **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. |

#### ST\_LayoutTarget (Layout Target)

This simple type specifies the possible ways to layout the plot area.

|  |  |
| --- | --- |
| **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. |

#### ST\_LblAlgn (Label Alignment)

This simple type specifies the possible ways to align the tick labels.

|  |  |
| --- | --- |
| **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. |

#### ST\_LblOffset (Label Offset)

This simple type specifies that its contents contain a percentage of the default value, between 0% and 1000%.

#### ST\_LegendPos (Legend Position)

This simple type specifies the possible positions for a legend.

|  |  |
| --- | --- |
| **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. |

#### ST\_LogBase (Logarithmic Base)

This simple type specifies that its contents contain a floating point number greater than or equal to two.

#### 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.

#### 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.

|  |  |
| --- | --- |
| **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. |

#### 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.

|  |  |
| --- | --- |
| **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. |

#### 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.

#### ST\_Orientation (Orientation)

This simple type specifies the possible ways to place a picture on a data point, series, wall, or floor.

|  |  |
| --- | --- |
| **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. |

#### ST\_Overlap (Overlap)

This simple type specifies that its contents contain a percentage between -100% and 100%.

#### ST\_PageSetupOrientation (Printed Page Orientation)

This simple type specifies the page orientation of the printed page(s) on which this chart shall appear.

|  |  |
| --- | --- |
| **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. |

#### ST\_Period (Period)

This simple type specifies that its contents contain an integer greater than or equal to 2.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| auto (Auto) | Specifies an application-specific marker shall be drawn at each data point. |

#### ST\_Perspective (Perspective)

This simple type specifies that its contents contain an integer between 0 and 100, whose contents are a percentage.

#### ST\_PictureFormat (Picture Format)

This simple type specifies the possible ways to place a picture on a data point, series, wall, or floor.

|  |  |
| --- | --- |
| **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. |

#### ST\_PictureStackUnit (Picture Stack Unit)

This simple type specifies that its contents contain a floating point number greater than zero.

#### ST\_RadarStyle (Radar Style)

This simple type specifies the possible styles of radar chart.

|  |  |
| --- | --- |
| **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. |

#### 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.

#### 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.

#### ST\_ScatterStyle (Scatter Style)

This simple type specifies the possible styles of scatter chart.

|  |  |
| --- | --- |
| **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. |

#### 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.

#### ST\_Shape (Shape)

This simple type specifies the possible shapes for a 3-D data marker.

|  |  |
| --- | --- |
| **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. |

#### ST\_SizeRepresents (Size Represents)

This simple type specifies the possible ways to represent data as bubble chart sizes.

|  |  |
| --- | --- |
| **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. |

#### ST\_Skip (Skip)

This simple type specifies that its contents contain an integer greater than or equal to one.

#### ST\_SplitType (Split Type)

This simple type specifies the possible ways to split a pie of pie or bar of pie chart.

|  |  |
| --- | --- |
| **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 |

#### 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.

#### ST\_TickLblPos (Tick Label Position)

This simple type specifies the possible positions for tick labels.

|  |  |
| --- | --- |
| **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. |

#### ST\_TickMark (Tick Mark)

This simple type specifies the possible positions for tick marks.

|  |  |
| --- | --- |
| **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. |

#### ST\_TimeUnit (Time Unit)

This simple type specifies a unit of time.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| days (Days) | Specifies the chart data shall be shown in days. |
| months (Months) | Specifies the chart data shall be shown in months. |
| years (Years) | Specifies the chart data shall be shown in years. |

#### ST\_TrendlineType (Trendline Type)

This simple type specifies all styles of trendline which are available for series in a chart.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| exp (Exponential) | Specifies the trendline shall be an exponential curve in the form y = abx. |
| 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 = ax6 + bx5+cx4 + dx3 + ex2 + fx + g. |
| power (Power) | Specifies the trendline shall be a power curve in the form y = axb. |

#### ST\_DepthPercentWithSymbol (Depth Percent with Symbol)

This simple type specifies that its contents contain a percentage between 20% and 2000%.

#### ST\_HPercentWithSymbol (Height Percent with Symbol)

This simple type specifies that its contents contain a percentage between 5% and 500%.

#### ST\_GapAmountPercent (Gap Amount Percentage)

This simple type specifies that its contents contain a percentage between 0% and 500%.

##### ST\_SecondPieSizePercent (Second Pie Size Percentage)

This simple type specifies that its contents contain a percentage between 5% and 200%.

##### ST\_HoleSizePercent (Hole Size Percentage)

This simple type specifies that its contents contain a percentage between 1% and 90%.

##### ST\_LblOffsetPercent (Label Offset Percentage)

This simple type specifies that its contents contain a percentage between 0% and 1000%.

##### ST\_OverlapPercent (Overlap Percentage)

This simple type specifies that its contents contain a percentage between -100% and 100%.

##### ST\_BubbleScalePercent (Bubble Scale Percentage)

This simple type specifies that its contents contain a percentage between 0% and 300%.

##### ST\_Thickness (Thickness Percentage)

This simple type specifies that its contents contain a percentage.

##### ST\_ThicknessPercent (Thickness Percentage)

This simple type specifies that its contents contain a percentage.

## 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.

### Elements

The following element define the contents of the ChartDrawing namespace:

#### 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.

#### 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.

|  |  |
| --- | --- |
| **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. |
| rotWithShape | 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. |

#### 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.

#### 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.

#### 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.

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| preferRelativeResi ze (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 |
| Preferred) | relative to the original picture size as opposed to the current picture size. |

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| descr (Alternative | Specifies alternative text for the current DrawingML object, for use by assistive technologies or applications which do not display the current object. |
| 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 application can have settings which allow this object to be viewed. *end note*] |
| id (Unique | 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. |
| 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*] |
| title (Title) | Specifies the title (caption) of the current DrawingML object. |

#### 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.

|  |  |  |
| --- | --- | --- |
| **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. | |
| **Attributes** | | **Description** |
| http://purl.oclc.or g/ooxml/drawing ml/main | |  |

#### 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.

|  |  |
| --- | --- |
| **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. |
| 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*] |

#### 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) | 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). |
| 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). |

#### 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:

#### 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. |
| 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*] |

#### 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.

#### 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 | 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. |

#### nvCxnSpPr (Connector Non Visual Properties)

This element specifies all non-visual properties for a connection shape. This element is a container for the nonvisual 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.

#### 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.

#### 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.

#### 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.

#### 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.

#### pic (Picture)

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

|  |  |
| --- | --- |
| **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. |
| 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*] |

#### 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.

#### 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 nonvisual 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. | |
| 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. |
| 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*] |
| 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. |

#### 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 | 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. |

#### 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.

#### 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.

#### 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.

#### 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.

#### xfrm (Graphic Frame Transform)

This element specifies a 2-D transform to be applied to a Graphic Frame.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| flipH (Horizontal | Specifies a horizontal flip. When true, this attribute defines that the shape is flipped horizontally about the center of its bounding box. |
| 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. |
| 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. |

#### 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.

### Simple Types

This is the complete list of simple types dedicated to DrawingML – Chart Drawings.

#### 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.

## DrawingML - Diagrams

A DrawingML diagram allows the definition of diagrams using DrawingML objects and constructs. This namespace defines the contents of a DrawingML diagram.

### Diagram Definition

This section specifies the elements which define the layout and hierarchy of a diagram based on its constituent nodes and connections.

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| idx (Adjust Handle Index) | Adjust value index. Different shapes support different adjust handles. |
| val (Value) | An absolute value. |

#### adjLst (Shape Adjust List)

This element is simply a list of shape adjusts.

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| rev (Revision Number) | The revision number of an algorithm. |
| type (Algorithm Type) | Specifies the algorithm type. |

#### cat (Category)

This element specifies a category in the user interface where this layout definition displays to the user.

|  |  |
| --- | --- |
| **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. |
| type (Category Type) | Specifies the category type associated with the element. |

#### catLst (Category List)

This element is simply a list of cat elements.

#### choose (Choose Element)

The choose element wraps if/else blocks into a choose block.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| name (Name) | A unique name associated with the choose statement. |

#### 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.

|  |  |
| --- | --- |
| **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. |

#### constr (Constraint)

This element is used to specify size, position of nodes, text values, and layout dependencies between nodes in a layout definition.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| fact (Factor) | Factor used in a reference constraint or a rule in order to modify a referenced value by the factor defined. |
| for (For) | Specifies the axis of layout nodes to apply a constraint or rule to. |
| forName (For Name) | Specifies the name of the layout node to apply a constraint or rule to. |
| op (Operator) | The operator constraint used to evaluate the condition. |
| ptType (Data Point Type) | Specifies the type of data point to select. |
| refFor (Reference For) | The for value of the referenced constraint. |
| refForName | The name of the layout node referenced by a reference constraint. |
| refPtType (Reference Point | The point type used int he referenced constraint. |
| refType (Reference Type) | Specifies the type of a reference constraint. |
| type (Constraint Type) | Specifies the constraint to apply to this layout node. |
| val (Value) | Specifies an absolute value instead of reference another constraint. |

#### constrLst (Constraint List)

This element is simply a list of constraints.

#### dataModel (Data Model)

The data for this instance of the diagram. Either a sample data model, or the data the user has entered.

#### 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. |
| val (Value) | The string which is used as the description of the layout definition. |

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| name (Name) | A unique name associated with the choose statement. |

#### extLst (Extension List)

This element specifies an extension list, within which all future extensions are defined within ext elements.

#### 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.

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Description** | |
| axis (Axis) | Specifies the axis on which to select data from the data model. | |
| cnt (Count) | Specifies the count of items to use in a data set. |
| hideLastTrans | 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. |
| name (Name) | A unique identifier for the layout node. |
| ptType (Data Point Type) | Specifies the type of data point to select. |
| ref (Reference) | When used on a for-each element, causes the specified for-each element to be used instead. |
| st (Start) | Specifies where to start in a data set. |
| 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. |

#### if (If)

Like an if statement in a programming language, wraps elements which are to be used under the conditions defined by its attributes.

|  |  |  |
| --- | --- | --- |
| **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". | |
| axis (Axis) | Specifies the axis on which to select data from the data model. | |
| cnt (Count) | Specifies the count of items to use in a data set. | |
| func (Function) | The function used to evaluate the if condition. | |
| name (Name) | A unique identifier for the layout node. |
| op (Operator) | The operator used to evaluate the condition. |
| ptType (Data Point Type) | Specifies the type of data point to select. |
| st (Start) | Specifies where to start in a data set. |
| 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. |
| val (Value) | An absolute value. |

#### 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. |
| minVer (Minimum Version) | Minimum product version that can support this layout definition. |
| uniqueId (Unique | The unique identifier for this layout definition. |

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| defStyle (Default Style) | This attribute defines a reference to a default style which is to be applied to the diagram. |
| minVer (Minimum Version) | Minimum product version that can support this Diagram Layout. |
| resId (Resource Identifier) | Resource ID used internally. |
| uniqueId (Unique | The unique identifier for this layout definition. |

#### 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.

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| chOrder (Child Order) | Specifes the ordering of the child layout nodes for a given layout node. |
| moveWith (Move With) | Reference to another layout node that this layout node moves with. |
| name (Name) | A unique identifier for the layout node. |
| styleLbl (Style Label) | Specify which formatting option from a style or color variation should be applied to this layout node. |

#### param (Parameter)

The parameter element modifies the default behavior of an algorithm.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| type (Parameter Type) | Specifies the parameter which is being modified. |
| val (Value) | Specifies the actual value to be given to the parameter type defined by the type attribute. |

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| axis (Axis) | Specifies the axis on which to select data from the data model. |
| cnt (Count) | Specifies the count of items to use in a data set. |
| hideLastTrans | 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. |
| ptType (Data Point Type) | Specifies the type of data point to select. |
| st (Start) | Specifies where to start in a data set. |
| 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. |

#### 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:

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| cs (Explicit | Specifies the relationship ID for the explicit relationship to the Diagram Colors part used by this diagram. |
| dm (Explicit Relationship to | Specifies the relationship ID for the explicit relationship to the Diagram Data part used by this diagram. |
| Namespace: | http://schemas.openxmlformats.org/officeDocument/2006/relationships/diagramD ata or the document shall be considered non-conformant. |
| lo (Explicit | Specifies the relationship ID for the explicit relationship to the Diagram Layout Definition part used by this diagram. |
| qs (Explicit | Specifies the relationship ID for the explicit relationship to the Diagram Style part used by this diagram. |

#### 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. |

#### rule (Rule)

This element allows for a rule to be specified which changes the value of an existing constraint.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| fact (Factor) | Factor used in a reference constraint or a rule in order to modify a referenced value by the factor defined. |
| for (For) | Specifies the axis of layout nodes to apply a constraint or rule to. |
| forName (For Name) | Specifies the name of the layout node to apply a constraint or rule to. |
| max (Max Value) | Sets the maximum value for a constraint so rules can no longer increase the constraint beyond that value. |
| ptType (Data Point Type) | Specifies the type of data point to select. |
| type (Constraint Type) | Specifies the constraint to apply to this layout node. |
| val (Value) | Specifies an absolute value instead of reference another constraint. |

#### ruleLst (Rule List)

This element is simply a list of rules.

#### 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.

|  |  |
| --- | --- |
| **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. |

#### shape (Shape)

The shape displayed by the containing layout node. Not all layout nodes display shapes.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| blip (Relationship to | Specifies the relationship ID of the explicit relationship to an image which shall be used as the image for the contents of this shape. |
| blipPhldr (Image Placeholder) | Specifies whether to use an image placeholder or not. |
| hideGeom (Hide Geometry) | When set to "true", hides the geometry of the shape. The text is still visible. |
| lkTxEntry (Prevent Text Editing) | Prevents text editing on this shape. |
| rot (Rotation) | Rotates the shape by the specified number of degrees. |
| type (Shape Type) | Specifies the type of shape. |
| 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. |

#### style (Shape Style)

This element specifies the style information for a shape, as defined by its DrawingML child elements.

#### 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.

|  |  |
| --- | --- |
| **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. |

#### title (Title)

Title of the Diagram Layout.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| lang (Language) | Specifies the language of the title or description of this layout definition. |
| val (Value) | Specifies the title or description of this layout definition. |

#### varLst (Variable List)

This element consists of a list of variables which interact with user interface components.

### Data

This section specifies the data that is to be contained within a diagram.

#### 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.

#### cxn (Connection)

This element defines a connection between two points. A connection defines a relationship between two points in a diagram.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| destId (Destination Identifier) | The model identifier of the destination point for a connection. |
| destOrd (Destination | The relative position of the destination point among it's siblings. |
| modelId (Model | The unique identifier associated with this cxn. |
| parTransId (Parent Transition | 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, |
| presId (Presentation | 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). |
| sibTransId (Sibling | 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*] |
| srcId (Source Identifier) | The model identifier of the source point for a connection. |
| srcOrd (Source Position) | The relative position of the source point among it's siblings. |
| type (Point Type) | The type of point, which corresponds to a connection in this case. |

#### 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.

#### 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:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | | **Description** | | | |
| coherent3DOff | | Enables or disables the Coherent 3D behavior for styles that specify this property. | | | |
| csCatId (Color | | This attribute specifies the identifier of the current color transform category. | | | |
| csTypeId (Color | | This attribute specifies the identifier of the currently applied color transform. | | | |
| custFlipHor (Custom Horizontal | | | Specifies if there is a custom horizontal flip applied. | | |
| custFlipVert (Custom Vertical | | | Specifies if there is a custom vertical flip applied. | | |
| custLinFactNeighb orX (Neighbor Offset Width) | | | Specifies the percentage of the neighbor's width used for offsetting shape. | | |
| custLinFactNeighb orY (Neighbor Offset Height) | | | Specifies the percentage of the neighbor's height used for offsetting shape. | | |
| custLinFactX (Custom Factor | | | Specifies the percentage of the current shape width used for offsetting the shape. | | |
| custLinFactY (Custom Factor | | | Specifies the percentage of the current shape height used for offsetting the shape. | | |
| custRadScaleInc | | | Specifies the amount that the include angle has been scaled by. | | |
| custRadScaleRad (Radius Scale) | | | Specifies how much the radius has been scaled. | | |
| custScaleX (Width Scale) | | | Specifies the amount that the width has been scaled by. | | |
| custScaleY (Height Scale) | | | Specifies the amount that the height has been scaled by. | | |
| custSzX (Fixed Width Override) | | Specifies a fixed width override for a shape. | |
| custSzY (Fixed Height Override) | | Specifies a fixed height override for a shape. | |
| custT (Text Changed) | | Specifies if the text has been customized which allows layout to ignore automatic formatting options available to the text. | |
| loCatId (Current Diagram Category) | | Specifies the current identifier of the layout category applied to the diagram. | |
| loTypeId (Current Diagram Type) | | Specifies the identifier for the layout currently applied to the diagram. | |
| phldr (Placeholder) | | Indicates that the point is a placeholder or sample item. | |
| 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. | |
| presAssocID | | The point associated with this presentation element. This identifier is used together with presName to create a unique key for presentation element indexing. | |
| presName | | The layout node name of this presentation element. This name is used together with presAssocID to create a unique key for presentation element indexing. | |
| presStyleCnt (Presentation Style | | Specifies the layout node style count of this presentation element. | |
| presStyleIdx (Presentation Style | | Specifies the layout node style index of this presentation element. | | | | |
| presStyleLbl (Presentation Style | Specifies the layout node style label of this presentation element. | | | |
| qsCatId (Current Style Category) | Specifies the identifier of the category of the currently applied quick style. | | | |
| qsTypeId (Current Style Type) | Specifies the identifier of the currently applied quick style. | | | |

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| cxnId (Connection Identifier) | The model identifier of the connection that represents the transition node. |
| modelId (Model | The unique identifier of the element within the data model. This identifier should be unique only relative to the containing data model. |
| type (Point Type) | The type of point. |

#### ptLst (Point List)

This element simply holds a list of points within the data model.

#### 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 | 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. |

#### 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.

#### whole (Whole E2O Formatting)

Formatting that applies to the entire diagram object, and not just the background, includes line and effect properties.

### Color Information

This section defines the coloring information that is to be associated with a diagram.

#### cat (Color Transform Category)

This element specifies the category in the user interface that a color transform is to be displayed within.

|  |  |
| --- | --- |
| **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. |
| type (Category Type) | The category type used to organize the color transforms in the user interface. |

#### 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.

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| minVer (Minimum  Version) | The minimum product version that can support this color transform. |
| uniqueId (Unique | A unique id associated with the color transform definition. |

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| minVer (Minimum Version) | The minimum product version that can support the associated color transform definition. |
| resId (Resource ID) | This attribute is the id which associates this header to the actual color transform definition. |
| uniqueId (Unique | This attribute defines a unique identifier for the associated color transform definition. |

#### 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.

#### 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. |
| val (Description Value) | The string which is used as the description of the color transform definition. |

#### effectClrLst (Effect Color List)

This element defines a list of colors applied to effects within a color transform.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| hueDir (Hue Direction) | The direction around the color wheel the hue shift (if defined) occurs. |
| meth (Color Application Method | The method used to apply the color transform. |

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| hueDir (Hue Direction) | The direction around the color wheel the hue shift (if defined) occurs. |
| meth (Color Application Method | The method used to apply the color transform. |

#### 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

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| hueDir (Hue Direction) | The direction around the color wheel the hue shift (if defined) occurs. |
| meth (Color Application Method | The method used to apply the color transform. |

#### 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.

|  |  |
| --- | --- |
| **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. |

#### 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. |
| val (Description Value) | A string used for a description of a color transform definition. |

#### 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

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| hueDir (Hue Direction) | The direction around the color wheel the hue shift (if defined) occurs. |
| meth (Color Application Method | The method used to apply the color transform. |

#### 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

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| hueDir (Hue Direction) | The direction around the color wheel the hue shift (if defined) occurs. |
| meth (Color Application Method | The method used to apply the color transform. |

#### 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

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| hueDir (Hue Direction) | The direction around the color wheel the hue shift (if defined) occurs. |
| meth (Color Application Method | The method used to apply the color transform. |

### Style Definitions

This section describes the styling information to be associated with a diagram.

#### cat (Category)

The category in the user interface where this quick style displays in the user interface.

|  |  |
| --- | --- |
| **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. |
| type (Category Type) | Category type. This is used to organize the quick style in the user interface. |

#### catLst (Category List)

This element is simply a list of categories.

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| lang (Natural Language) | The natural language of the title or description of this quick style. |
| val (Description Value) | The string used for the description. |

#### 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.

#### scene3d (3-D Scene)

The 3-D scene which consists of a camera, a light rig, and an optional backdrop to catch shadows.

#### sp3d (3-D Shape Properties)

A set of 3-D properties which a shape can contain.

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Description** | |
| contourW (Contour | Defines the width of the contour on the shape. | |
| extrusionH | | Defines the height of the extrusion applied to the shape. |
| prstMaterial (Preset Material | | Defines the preset material which is combined with the lighting properties to give the final look and feel of a shape. |
| z (Shape Depth) | | Defines the z coordinate for the 3D shape. |

#### styleDef (Style Definition)

This element is the root tag for a style definition.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| minVer (Minimum Version) | The minimum product version that can support this quick style. |
| uniqueId (Unique Style ID) | Unique ID that identifies a style. |

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| minVer (Minimum Version) | The minimum product version that can support this quick style. |
| resId (Resource ID) | This attribute is the id which associates this header to the actual style definition part. |
| uniqueId (Unique Style ID) | This attribute defines a unique identifier for the associated style definition. |

#### 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.

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| name (Style Name) | The name of the style. This apears as the tooltip in the user interface. |

#### title (Title)

This element defines the title given to a style definition header. The title is simply a name for the style definition.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| lang (Natural Language) | The natural language of the title or description of this quick style. |
| val (Description Value) | The string used for the description. |

#### txPr (Text Properties)

This element defines special text formatting that can be applied to text through a style label.

### Layout Definition

This section specifies the node layout information to be associated with a diagram.

#### animLvl (Level Animation)

This variable is used to indicate the animate by level string which is displayed to a user in the user interface.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Level | This attribute indicates the string to use for level animation in the user interface. |

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (One By One Animation Value) | Specifies the type of one-by-one animation to use for a diagram. |

#### bulletEnabled (Show Insert Node)

This element is used to indicate whether to enable user interface components associated with inserting a node in the data model.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Show Insert Node Value) | This attribute is used to indicatewhether a user interface for inserting a node should be enabled. A value of true indicates that the user interface should be enabled. |

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Maximum | 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. |

#### chPref (Preferred Number of Children)

This variable indicates the number of children that the current node prefers to have. [*Note*: For example, this could be used to guide how many shapes to add by default to a diagram at various levels in the hierarchy. *end note*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Preferred Number of CHildren | This attribute indicates the number of children that the current node prefers to have. A value of -1 indicates an infinite number of children. Default value is -1. |

#### 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. |

#### hierBranch (Organization Chart Branch Style)

This element defines the layout style of a branch in an organizational chart.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Organization Chart Branch Style | The value of this attribute indicates the layout style of a branch in an organization chart. The default value is std. |

#### 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.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Show | This attribute value specifies when to show the ‘Insert Assistant’ user interface control and the ‘Change Layout’ user interface for this diagram. |

### Simple Types

This is the complete list of simple types dedicated to DrawingML – Diagrams.

#### ST\_AlgorithmType (Algorithm Types)

Types of available algorithms.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| composite (Composite) | The composite algorithm specifies the size and position for all child layout nodes. You can use it to |
| 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 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. |

#### 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.

|  |  |
| --- | --- |
| **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. |

#### 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.

|  |  |
| --- | --- |
| **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". |

#### ST\_ArrowheadStyle (Arrowhead Styles)

This simple type defines different arrowhead style types for connectors.

|  |  |
| --- | --- |
| **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. |

#### 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.

|  |  |
| --- | --- |
| **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. |

#### ST\_AxisType (Axis Type)

This simple type defines different node sets in relation to the current context node.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| ancst (Ancestor) | Specifies a set of nodes between the current context node and the root node, including the root node. |
| 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. |
| ch (Child) | Specifies a set of children of the current context node. |
| des (Descendant) | Specifies a set of all nodes beneath the current context node. |
| desOrSelf (Descendant or Self) | Specifies a set of all nodes beneath the current context node, including the context node. |
| follow (Follow) | Specifies the set of nodes which are peers after the context node and all descendants of the peers. |
| followSib (Follow Sibling) | Specifies the set of nodes which are peers after the context node. |
| none (None) | Specifies no node. |
| par (Parent) | Specifies the parent node. |
| preced (Preceding) | Specifies the set of nodes which are peers before the context node and all the descendants of the peers. |
| precedSib (Preceding Sibling) | Specifies the set of nodes which are peers before the context node. |
| root (Root) | Specifies the top-most node of the diagram. |
| self (Self) | Specifies the calling context node. |

#### ST\_AxisTypes (Axis Type List)

This simple type represents a list of axis types.

#### ST\_BendPoint (Bend Point)

This simple type defines where a bend is to occur within a connection between two nodes.

|  |  |
| --- | --- |
| **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. |

#### ST\_Booleans (Boolean List.)

A list of booleans.

#### ST\_BoolOperator (Boolean Constraint)

This simple type specified Boolean operations which can be applied to compare constraints.

|  |  |
| --- | --- |
| **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 |

#### ST\_Breakpoint (Breakpoint)

This simple type defines at what point the wrapping of nodes occurs for the snake algorithm.

|  |  |
| --- | --- |
| **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. |

#### ST\_CenterShapeMapping (Center Shape Mapping)

This simple type defines the behavior of the cycle algorithm.

|  |  |
| --- | --- |
| **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. |

#### ST\_ChildAlignment (Child Alignment)

This simple type defines how to align a node in its allocated space.

|  |  |
| --- | --- |
| **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. |

#### ST\_ChildDirection (Child Direction)

This simple type defines the layout direction of child nodes related to a specific parent node.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| horz (Horizontal) | Specifies that the child nodes are to be laid out |
| vert (Vertical) | Specifies that the child nodes are to be laid out vertically. |

#### ST\_ChildOrderType (Child Order)

This simple type specifies the child order for a given layout node.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| b (Bottom) | Child order along the bottom. |
| t (Top) | Top child order. |

#### 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.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| cycle (Cycle) | The colors apply from A to B to A if A and B were the colors present. |
| repeat (Repeat) | The colors apply from A through B to A through B if A through B were the colors present. |
| span (Span) | The colors interpolate from A to B across the entire diagram if A and B were the colors present. |

#### ST\_ConnectorDimension (Connector Dimension)

This simple type defines the dimensionality of the connection between two nodes.

|  |  |
| --- | --- |
| **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. |

#### ST\_ConnectorPoint (Connector Point)

This simple type defines different connection sites available on a node.

|  |  |
| --- | --- |
| **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. |

#### 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.

|  |  |
| --- | --- |
| **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. |

#### ST\_ConstraintRelationship (Constraint Relationship)

This simple type specifies the types of constraint relationships which are present and can be used.

|  |  |
| --- | --- |
| **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. |

#### ST\_ConstraintType (Constraint Type)

This simple type defines the list of possible constraints available for use.

|  |  |
| --- | --- |
| **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. |
| 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. |
| 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. |

#### 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.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| revDir (Reverse Direction) | Specifies that the direction is to be revered on a subsequent row or column. |
| sameDir (Same Direction) | Specifies that the direction is to be maintained on a subsequent row or column. |

#### ST\_CxnType (Connection Type)

This simple type defines the different types of relationships that can be defined between two nodes.

|  |  |
| --- | --- |
| **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. |

#### ST\_DiagramHorizontalAlignment (Horizontal Alignment)

This simple type defines the horizontal alignment.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| ctr (Center) | Specifies center alignment. |
| l (Left) | Specifies left alignment. |
| none (None) | Specifies no alignment defined. |
| r (Right) | Specifies right alignment. |

#### ST\_DiagramTextAlignment (Text Alignment)

This simple type defines alignment types for text within a node.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| ctr (Center) | Specifies center aligned text. |
| l (Left) | Specifies left aligned text. |
| r (Right) | Specifies right aligned text. |

#### ST\_Direction (Diagram Direction Definition)

This simple type defines the possible values for a diagram's direction when displayed in an application.

|  |  |
| --- | --- |
| **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. |

#### ST\_ElementType (Data Point Type)

This simple type defines the different types of data points which are supported.

|  |  |
| --- | --- |
| **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. |

#### ST\_ElementTypes (Diagream Layout Node Type List)

A list of diagram layout node types.

#### ST\_FallbackDimension (Fallback Dimension)

Specifies the dimensionality by which nodes can grow or shrink automatically.

|  |  |
| --- | --- |
| **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. |

#### ST\_FlowDirection (Flow Direction)

This simple type defines how the progression of new nodes are to be entered into the diagram.

|  |  |
| --- | --- |
| **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. |

#### ST\_FunctionArgument (Function Argument)

Conditional expression function argument.

#### ST\_FunctionOperator (Function Operator)

This simple type defines the condition expression functions which can be used to perform operations.

|  |  |
| --- | --- |
| **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. |

#### ST\_FunctionType (Function Type)

This simple type defines the set of available conditional expression function types present for use.

|  |  |
| --- | --- |
| **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 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. |

#### ST\_FunctionValue (Function Value)

Conditional expression function value.

#### ST\_GrowDirection (Grow Direction)

This simple type defines different starting locations for nodes within the snake algorithm.

|  |  |
| --- | --- |
| **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. |

#### 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.

|  |  |
| --- | --- |
| **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 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. |

#### ST\_HierBranchStyle (Hierarchy Branch Style Definition)

This simple type specifies the possible values for the branch style of a hierarchy diagram.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| hang (Hanging) | The branch style is hanging from the parent. |
| init (Initial) | This means that the value has not been set. |
| l (Left) | The branch style falls off the left. |
| r (Right) | The branch style falls off the right. |
| std (Standard) | The standard branch style is to be used. |

#### 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.

|  |  |
| --- | --- |
| **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. |

#### ST\_Index1 (1-Based Index)

A 1-based index.

#### ST\_Ints (Integer List)

A list of integers.

#### ST\_LayoutShapeType (Layout Shape Type)

All of the available shape types.

#### ST\_LinearDirection (Linear Direction)

This simple type defines the direction of growth of new nodes.

|  |  |
| --- | --- |
| **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 |

#### ST\_ModelId (Model Identifier)

The unique ID of the element within the data model. Model Identifiers can be either longs or guids.

#### 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.

#### ST\_NodeHorizontalAlignment (Node Horizontal Alignment)

This simple type defines the horizontal alignment of a node.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| ctr (Center) | Specifies center alignment. |
| l (Left) | Specifies left alignment. |
| r (Right) | Specifies right alignment. |

#### ST\_NodeVerticalAlignment (Node Vertical Alignment)

This simple type defines the vertical alignment of a node.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| b (Bottom) | Specifies bottom alignment. |
| mid (Middle) | Specifies middle alignment. |
| t (Top) | Specifies top alignment. |

#### 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.

|  |  |
| --- | --- |
| **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. |

#### ST\_OutputShapeType (Output Shape Type)

Shapes which are special specifically for a DrawingML diagram.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| conn (Connection) | Connection shape type. |
| none (None) | None. |

#### 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.

|  |  |  |
| --- | --- | --- |
| **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 | |
| 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. | |
| 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. | |
| 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. | |

#### ST\_ParameterVal (Parameter Values)

Specifies the list of parameter types that can be used by a diagram.

#### ST\_PtType (Point Type)

This simple type defines the different point types which can be utilized to create diagrams in DiagramML.

|  |  |
| --- | --- |
| **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. |

#### ST\_PyramidAccentPosition (Pyramid Accent Position)

This simple type defines different positioning for the accent shapes which can be associated with a pyramid algorithm.

|  |  |
| --- | --- |
| **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. |

#### ST\_PyramidAccentTextMargin (Pyramid Accent Text Margin)

This simple type defines different ways to lay out text in the accent shape for a pyramid algorithm.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| stack (Stack) | Specifies that all accent shape text is to be left aligned. |
| step (Step) | Specifies that all accent shape text is to be relative to the pyramid. |

#### 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.

|  |  |
| --- | --- |
| **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. |

#### ST\_RotationPath (Rotation Path)

This simple type defines rotation properties for nodes within the cycle algorithm.

|  |  |
| --- | --- |
| **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. |

#### ST\_SecondaryChildAlignment (Secondary Child Alignment)

This simple type defines different alignment properties of the both hanging layout type of the hierarchy algorithm.

|  |  |
| --- | --- |
| **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. |

#### ST\_SecondaryLinearDirection (Secondary Linear Direction)

This simple type defines different directions for the nodes in a both hanging layout in the hierarchy algorithm.

|  |  |
| --- | --- |
| **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. |

#### ST\_StartingElement (Starting Element)

This simple type defines behavior for the first node in a cycle algorithm.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| node (Node) | Specifies that a node should be placed first. |
| trans (Transition) | Specifies that a transition should be placed first. |

#### ST\_TextAnchorHorizontal (Text Anchor Horizontal)

This simple type defines horizontal anchor points for text.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| ctr (Center) | Specifies text to be anchored to the center. |
| none (None) | Specifies no horizontal text anchor. |

#### ST\_TextAnchorVertical (Text Anchor Vertical)

This simple type defines vertical anchor points for text.

|  |  |
| --- | --- |
| **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. |

#### ST\_TextBlockDirection (Text Block Direction)

This simple type defines different layout options for text within a node.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| horz (Horizontal) | Specifies that the text is to be horizontal. |
| vert (Vertical Direction) | Specifies that the text is to be vertical. |

#### ST\_TextDirection (Text Direction)

This simple type defines different way the growth of additional text can occur within a node.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| fromB (From Bottom) | Specifies additional text grows from the bottom. |
| fromT (From Top) | Specifies additional text grows from the top. |

#### ST\_UnsignedInts (Unsigned Integer List)

A list of unsigned integers.

#### ST\_VariableType (Variable Type)

Conditional expression variable type.

|  |  |
| --- | --- |
| **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. |
| 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. |

#### ST\_VerticalAlignment (Vertical Alignment)

This simple type defines different vertical alignment parameters.

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| b (Bottom) | Specifies bottom aligned. |
| mid (Middle) | Specifies middle aligned. |
| none (None) | Specifies no vertical alignment. |
| t (Top) | Specifies top aligned. |

#### ST\_PrSetCustVal (Property Set Customized Value)

This simple type defines customization percentage values for certain elements in DrawingML.