## Math

The following documentation specifies the XML representation of mathematical text for OOXML. This shared ML is known as that Office Math Markup Language (OMML). Mathematical text represented by OMML includes but is not limited to: equations, expressions, formulas, matrices and other mathematical elements. The outermost OMML element of an instance of mathematical text in display mode is oMathPara, a math paragraph of one or more instances of mathematical text. Each instance of mathematical text inside the math paragraph is represented as a single oMath. Inside each oMath is a combination of mathematical runs (r) and objects or functions such as accents (acc) or fractions(f).

### Table of Contents

**This subclause is informative.**

[22.1. Math 22-1](#_Toc132650782)

[22.1.1. Table of Contents 22-1](#_Toc132650783)

[22.1.2. Elements 22-5](#_Toc132650784)

[22.1.2.1. acc (Accent) 22-5](#_Toc132650785)

[22.1.2.2. accPr (Accent Properties) 22-5](#_Toc132650786)

[22.1.2.3. aln (Alignment) 22-6](#_Toc132650787)

[22.1.2.4. alnScr (Align Scripts) 22-7](#_Toc132650788)

[22.1.2.5. argPr (Argument Properties) 22-8](#_Toc132650789)

[22.1.2.6. argSz (Argument Size) 22-9](#_Toc132650790)

[22.1.2.7. bar (Bar) 22-10](#_Toc132650791)

[22.1.2.8. barPr (Bar Properties) 22-11](#_Toc132650792)

[22.1.2.9. baseJc (Matrix Base Justification) 22-11](#_Toc132650793)

[22.1.2.10. begChr (Delimiter Beginning Character) 22-13](#_Toc132650794)

[22.1.2.11. borderBox (Border-Box Object) 22-14](#_Toc132650795)

[22.1.2.12. borderBoxPr (Border-Box Properties) 22-14](#_Toc132650796)

[22.1.2.13. box (Box Object) 22-15](#_Toc132650797)

[22.1.2.14. boxPr (Box Properties) 22-15](#_Toc132650798)

[22.1.2.15. brk (Break) 22-16](#_Toc132650799)

[22.1.2.16. brkBin (Break on Binary Operators) 22-17](#_Toc132650800)

[22.1.2.17. brkBinSub (Break on Binary Subtraction) 22-18](#_Toc132650801)

[22.1.2.18. cGp (Matrix Column Gap) 22-19](#_Toc132650802)

[22.1.2.19. cGpRule (Matrix Column Gap Rule) 22-20](#_Toc132650803)

[22.1.2.20. chr (Character) 22-21](#_Toc132650804)

[22.1.2.21. count (Matrix Column Count) 22-22](#_Toc132650805)

[22.1.2.22. cSp (Minimum Matrix Column Width) 22-23](#_Toc132650806)

[22.1.2.23. ctrlPr (Control Properties) 22-24](#_Toc132650807)

[22.1.2.24. d (Delimiter Object) 22-25](#_Toc132650808)

[22.1.2.25. defJc (Default Justification) 22-26](#_Toc132650809)

[22.1.2.26. deg (Degree) 22-26](#_Toc132650810)

[22.1.2.27. degHide (Hide Degree) 22-27](#_Toc132650811)

[22.1.2.28. den (Denominator) 22-28](#_Toc132650812)

[22.1.2.29. diff (Differential) 22-28](#_Toc132650813)

[22.1.2.30. dispDef (Use Display Math Defaults) 22-29](#_Toc132650814)

[22.1.2.31. dPr (Delimiter Properties) 22-30](#_Toc132650815)

[22.1.2.32. e (Element (Argument)) 22-30](#_Toc132650816)

[22.1.2.33. endChr (Delimiter Ending Character) 22-32](#_Toc132650817)

[22.1.2.34. eqArr (Array Object) 22-32](#_Toc132650818)

[22.1.2.35. eqArrPr (Array Properties) 22-34](#_Toc132650819)

[22.1.2.36. f (Fraction Object) 22-34](#_Toc132650820)

[22.1.2.37. fName (Function Name) 22-35](#_Toc132650821)

[22.1.2.38. fPr (Fraction Properties) 22-35](#_Toc132650822)

[22.1.2.39. func (Function Apply Object) 22-36](#_Toc132650823)

[22.1.2.40. funcPr (Function Properties) 22-37](#_Toc132650824)

[22.1.2.41. groupChr (Group-Character Object) 22-37](#_Toc132650825)

[22.1.2.42. groupChrPr (Group-Character Properties) 22-37](#_Toc132650826)

[22.1.2.43. grow (n-ary Grow) 22-38](#_Toc132650827)

[22.1.2.44. hideBot (Hide Bottom Edge) 22-39](#_Toc132650828)

[22.1.2.45. hideLeft (Hide Left Edge) 22-40](#_Toc132650829)

[22.1.2.46. hideRight (Hide Right Edge) 22-40](#_Toc132650830)

[22.1.2.47. hideTop (Hide Top Edge) 22-41](#_Toc132650831)

[22.1.2.48. interSp (Inter-Equation Spacing) 22-42](#_Toc132650832)

[22.1.2.49. intLim (Integral Limit Locations) 22-42](#_Toc132650833)

[22.1.2.50. intraSp (Intra-Equation Spacing) 22-43](#_Toc132650834)

[22.1.2.51. jc (Justification) 22-43](#_Toc132650835)

[22.1.2.52. lim (Limit) 22-44](#_Toc132650836)

[22.1.2.53. limLoc (n-ary Limit Location) 22-44](#_Toc132650837)

[22.1.2.54. limLow (Lower-Limit Object) 22-45](#_Toc132650838)

[22.1.2.55. limLowPr (Lower-Limit Properties) 22-46](#_Toc132650839)

[22.1.2.56. limUpp (Upper-Limit Object) 22-46](#_Toc132650840)

[22.1.2.57. limUppPr (Upper-Limit Properties) 22-47](#_Toc132650841)

[22.1.2.58. lit (Literal) 22-47](#_Toc132650842)

[22.1.2.59. lMargin (Left Margin) 22-48](#_Toc132650843)

[22.1.2.60. m (Matrix Object) 22-49](#_Toc132650844)

[22.1.2.61. mathFont (Math Font) 22-51](#_Toc132650845)

[22.1.2.62. mathPr (Math Properties) 22-51](#_Toc132650846)

[22.1.2.63. maxDist (Maximum Distribution) 22-52](#_Toc132650847)

[22.1.2.64. mc (Matrix Column) 22-53](#_Toc132650848)

[22.1.2.65. mcJc (Matrix Column Justification) 22-54](#_Toc132650849)

[22.1.2.66. mcPr (Matrix Column Properties) 22-56](#_Toc132650850)

[22.1.2.67. mcs (Matrix Columns) 22-57](#_Toc132650851)

[22.1.2.68. mPr (Matrix Properties) 22-58](#_Toc132650852)

[22.1.2.69. mr (Matrix Row) 22-59](#_Toc132650853)

[22.1.2.70. nary (n-ary Operator Object) 22-60](#_Toc132650854)

[22.1.2.71. naryLim (n-ary Limit Location) 22-61](#_Toc132650855)

[22.1.2.72. naryPr (n-ary Properties) 22-62](#_Toc132650856)

[22.1.2.73. noBreak (No Break) 22-63](#_Toc132650857)

[22.1.2.74. nor (Normal Text) 22-63](#_Toc132650858)

[22.1.2.75. num (Numerator) 22-64](#_Toc132650859)

[22.1.2.76. objDist (Object Distribution) 22-65](#_Toc132650860)

[22.1.2.77. oMath (Office Math) 22-66](#_Toc132650861)

[22.1.2.78. oMathPara (Office Math Paragraph) 22-66](#_Toc132650862)

[22.1.2.79. oMathParaPr (Office Math Paragraph Properties) 22-67](#_Toc132650863)

[22.1.2.80. opEmu (Operator Emulator) 22-68](#_Toc132650864)

[22.1.2.81. phant (Phantom Object) 22-69](#_Toc132650865)

[22.1.2.82. phantPr (Phantom Properties) 22-69](#_Toc132650866)

[22.1.2.83. plcHide (Hide Placeholders (Matrix)) 22-69](#_Toc132650867)

[22.1.2.84. pos (Position) 22-70](#_Toc132650868)

[22.1.2.85. postSp (Post-Paragraph Spacing) 22-71](#_Toc132650869)

[22.1.2.86. preSp (Pre-Paragraph Spacing) 22-71](#_Toc132650870)

[22.1.2.87. r (Run) 22-72](#_Toc132650871)

[22.1.2.88. rad (Radical Object) 22-72](#_Toc132650872)

[22.1.2.89. radPr (Radical Properties) 22-73](#_Toc132650873)

[22.1.2.90. rMargin (Right Margin) 22-73](#_Toc132650874)

[22.1.2.91. rPr (Run Properties) 22-74](#_Toc132650875)

[22.1.2.92. rSp (Row Spacing (Array)) 22-74](#_Toc132650876)

[22.1.2.93. rSpRule (Row Spacing Rule) 22-75](#_Toc132650877)

[22.1.2.94. scr (Script) 22-77](#_Toc132650878)

[22.1.2.95. sepChr (Delimiter Separator Character) 22-77](#_Toc132650879)

[22.1.2.96. show (Phantom Show) 22-78](#_Toc132650880)

[22.1.2.97. shp (Shape (Delimiters)) 22-79](#_Toc132650881)

[22.1.2.98. smallFrac (Small Fraction) 22-79](#_Toc132650882)

[22.1.2.99. sPre (Pre-Sub-Superscript Object) 22-80](#_Toc132650883)

[22.1.2.100. sPrePr (Pre-Sub-Superscript Properties) 22-80](#_Toc132650884)

[22.1.2.101. sSub (Subscript Object) 22-81](#_Toc132650885)

[22.1.2.102. sSubPr (Subscript Properties) 22-81](#_Toc132650886)

[22.1.2.103. sSubSup (Sub-Superscript Object) 22-82](#_Toc132650887)

[22.1.2.104. sSubSupPr (Sub-Superscript Properties) 22-82](#_Toc132650888)

[22.1.2.105. sSup (Superscript Object) 22-83](#_Toc132650889)

[22.1.2.106. sSupPr (Superscript Properties) 22-83](#_Toc132650890)

[22.1.2.107. strikeBLTR (Border Box Strikethrough Bottom-Left to Top-Right) 22-84](#_Toc132650891)

[22.1.2.108. strikeH (Border Box Strikethrough Horizontal) 22-84](#_Toc132650892)

[22.1.2.109. strikeTLBR (Border Box Strikethrough Top-Left to Bottom-Right) 22-85](#_Toc132650893)

[22.1.2.110. strikeV (Border Box Strikethrough Vertical) 22-86](#_Toc132650894)

[22.1.2.111. sty (style) 22-87](#_Toc132650895)

[22.1.2.112. sub (Subscript (Pre-Sub-Superscript)) 22-87](#_Toc132650896)

[22.1.2.113. subHide (Hide Subscript (n-ary)) 22-88](#_Toc132650897)

[22.1.2.114. sup (Superscript (Superscript object)) 22-88](#_Toc132650898)

[22.1.2.115. supHide (Hide Superscript (n-ary)) 22-89](#_Toc132650899)

[22.1.2.116. t (Text) 22-89](#_Toc132650900)

[22.1.2.117. transp (Transparent (Phantom)) 22-90](#_Toc132650901)

[22.1.2.118. type (Fraction type) 22-91](#_Toc132650902)

[22.1.2.119. vertJc (Vertical Justification) 22-91](#_Toc132650903)

[22.1.2.120. wrapIndent (Wrap Indent) 22-93](#_Toc132650904)

[22.1.2.121. wrapRight (Wrap Right) 22-93](#_Toc132650905)

[22.1.2.122. zeroAsc (Phantom Zero Ascent) 22-94](#_Toc132650906)

[22.1.2.123. zeroDesc (Phantom Zero Descent) 22-95](#_Toc132650907)

[22.1.2.124. zeroWid (Phantom Zero Width) 22-96](#_Toc132650908)

[22.1.3. Simple Types 22-96](#_Toc132650909)

[22.1.3.1. ST\_BreakBin (Break Binary Operators) 22-96](#_Toc132650910)

[22.1.3.2. ST\_BreakBinSub (Break on Binary Subtraction) 22-97](#_Toc132650911)

[22.1.3.3. ST\_Char (Character) 22-97](#_Toc132650912)

[22.1.3.4. ST\_FType (Fraction Type) 22-98](#_Toc132650913)

[22.1.3.5. ST\_Integer2 (Integer value (-2 to 2)) 22-98](#_Toc132650914)

[22.1.3.6. ST\_Integer255 (Integer value (1 to 255)) 22-98](#_Toc132650915)

[22.1.3.7. ST\_Jc (Justification) 22-99](#_Toc132650916)

[22.1.3.8. ST\_LimLoc (Limit Location) 22-99](#_Toc132650917)

[22.1.3.9. ST\_Script (Script) 22-99](#_Toc132650918)

[22.1.3.10. ST\_Shp (Shape (Delimiters)) 22-100](#_Toc132650919)

[22.1.3.11. ST\_SpacingRule (Spacing Rule) 22-100](#_Toc132650920)

[22.1.3.12. ST\_Style (Style) 22-100](#_Toc132650921)

[22.1.3.13. ST\_TopBot (Top-Bottom) 22-101](#_Toc132650922)

[22.1.3.14. ST\_UnSignedInteger (Unsigned integer) 22-101](#_Toc132650923)

**End of informative text.**

### Elements

The following elements describe the contents of mathematical text.

#### acc (Accent)

This element specifies the accent function, consisting of a base and a combining diacritical mark. If accPr is omitted, the default accent is U+0302 (COMBINING CIRCUMFLEX ACCENT). [*Example*: Example accent functions are, and .

<m:acc>

<m:accPr>

<m:chr m:val="&#771;"/>

<m:ctrlPr/>

</m:accPr>

<m:e>

<m:r>

<m:t>a</m:t>

</m:r>

</m:e>

</m:acc>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_Acc) is located in §A.6.1. *End note*]

#### accPr (Accent Properties)

This element specifies the properties of the Accent function. If chr is omitted, the default accent character is U+0302 (COMBINING CIRCUMFLEX ACCENT). [*Example*: The diacritical mark ~ (tilde) is:

<m:accPr>

<m:chr m:val="&#771;"/>

</m:accPr>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_AccPr) is located in §A.6.1. *End note*]

#### aln (Alignment)

This element specifies the alignment property on the box object. It is utilized only when the box is designated as an operator emulator. When 1 or true, this operator emulator serves as an alignment point; that is, designated alignment points in other equations can be aligned with it.

When the element is absent, the default is for the parent structure to not be an alignment point.

When the element is present and the val attribute is absent, the default value of the val attribute is1meaning that this property’s parent structure is an alignment point.

[*Example*: For example, the following equation uses the operator emulator as an alignment point:

.

Its XML representation is as follows:

<m:oMath>

<m:r>

<m:t>a</m:t>

</m:r>

<m:box>

<m:boxPr>

<m:opEmu m:val="1"/>

<m:aln m:val="1"/>

<m:ctrlPr/>

</m:boxPr>

<m:e>

<m:r>

<m:t>==</m:t>

</m:r>

</m:e>

</m:box>

<m:r>

<m:t>b</m:t>

</m:r>

</m:oMath>

*End example*]

[*Example*:

Given the following equations:

Their associated XML representation is as follows:

<m:oMathPara>

<m:oMath>

<m:r>

<m:t>a+b</m:t>

</m:r>

<m:r>

<m:rPr>

<m:aln/>

</m:rPr>

<m:t>-c=2</m:t>

</m:r>

<m:r>

<m:rPr>

<m:sty m:val="p"/>

</m:rPr>

<w:br/>

</m:r>

</m:oMath>

<m:oMath>

<m:r>

<m:t>x</m:t>

</m:r>

<m:r>

<m:rPr>

<m:aln/>

</m:rPr>

<m:t>-y+z=3</m:t>

</m:r>

</m:oMath>

</m:oMathPara>

The subtraction symbols in each of the above equations have been identified as alignment points, so the equations are aligned at their subtraction symbols (because they are in the same Math paragraph). *End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### alnScr (Align Scripts)

This element specifies the alignment of scripts in the subscript/superscript function. When 1 or true, subscripts and superscripts are aligned to each other. When 0 or false, they are kerned to the shape of the base. If this element is omitted, scripts are not aligned. In other words, when the element is absent, the default is for the sub-superscript object to not align the superscript and subscript with each other.

When the element is present and the val attribute is absent, the default value of the val attribute is1meaning that this property is applied.

[*Example*: Example (OFF):; Example (ON):.

The XML representation of the second example above is:

<m:sSubSup>

<m:sSubSupPr>

<m:alnScr m:val="1"/>

</m:sSubSupPr>

<m:e>

<m:r>

<m:t>f</m:t>

</m:r>

</m:e>

<m:sub>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>2</m:t>

</m:r>

</m:sub>

<m:sup>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>3</m:t>

</m:r>

</m:sup>

</m:sSubSup>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### argPr (Argument Properties)

This element specifies any properties of the math argument. [*Example*: The XML below represents the argSz attribute on the base element of a box:

<m:box>

<m:boxPr>

<m:noBreak m:val="0"/>

<m:ctrlPr/>

</m:boxPr>

<m:e>

<m:argPr>

<m:argSz m:val="-1"/>

</m:argPr>

<m:r>

<m:t>abc</m:t>

</m:r>

</m:e>

</m:box>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OMathArgPr) is located in §A.6.1. *End note*]

#### argSz (Argument Size)

This element specifies the size, or script level, of an argument. If the element is omitted, the default argument size is 0.

Whether the element is absent or present without the val attribute, the default value of the val attribute is always 0. However, this property does not specify an absolute size, rather it’s absolute to its starting size. Most arguments begin at the normal size which will be defined as0in the list below, but some begin, by default smaller or larger than the normal size. For instance, 1 represents that it is one size larger than normal, 2 represents that it is two sizes larger than normal, -1 represents that it is one size smaller than normal and -2 represents that it is two sizes smaller than normal. The list below indicates the default starting sizes for each argument, if it is not in this list, then the size cannot be changed on that argument.

* box base: 0
* group-character base: 0
* lower limit: -1
* upper limit: -1
* n-ary operator subscript: -1
* n-ary operator superscript: -1
* radical degree: -2
* pre-sub-superscript subscript: -1
* pre-sub-superscript superscript: -1
* subscript: -1
* sub-superscript subscript: -1
* sub-superscript superscript: -1
* superscript: -1

[*Example*: The following example contains three runs: regular mathematical text, a box object with the base at script size(val=-1)and a box object with the base at script-script size(val=-2).

The XML below shows argSz used in the middlebox.

<m:box>

<m:boxPr>

<m:noBreak m:val="0"/>

</m:boxPr>

<m:e>

<m:argPr>

<m:argSz m:val="-1"/>

</m:argPr>

<m:r>

<m:t>abc</m:t>

</m:r>

</m:e>

</m:box>

Because the size is set to -1 on the degree argument, the degree argument is 1 size smaller normal.

*End example*]

|  |  |
| --- | --- |
| Attributes | Description |
| val (Value) | Specifies a value between -2 and 2 for the property defined by the parent XML element. The positive or negative sign specifies in which direction to change argument size; the absolute value specifies by how much.  The table below represents two cases in which argument size can be changed:  superscripts and boxes.  In the superscript object , by default the term 𝑐 has script-script size. Should the user wish for the 𝑐 to be shown at script size, val should be set to +1 (that is, one size larger). Should the user wish for 𝑐 to be shown at text size, val should be set to +2 (that is, two sizes larger).   |  |  | | --- | --- | | val of 𝒄 in | Display | | Default |  | | +1 |  | | +2 |  |  |  |  | | --- | --- | | val of 𝒂𝒃𝒄 | Display | | Default |  | | -1 |  | | -2 |  |   The possible values for this attribute are defined by the ST\_Integer2 simple type (§22.1.3.5). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_Integer2) is located in §A.6.1. *End note*]

#### bar (Bar)

This element specifies the bar function, consisting of a base argument and an overbar or underbar, as in and .

[*Example*: The XML below demonstrates the overbar in use.

<m:bar>

<m:barPr>

<m:pos m:val="top"/>

</m:barPr>

<m:e>

<m:r>

<m:t>a</m:t>

</m:r>

</m:e>

</m:bar>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_Bar) is located in §A.6.1. *End note*]

#### barPr (Bar Properties)

This element specifies properties of the bar function. If this element is omitted, the bar assumes its default location of top (the mathematical overbar).

[*Example*: (x with an underbar) is represented by the following XML representation:

<m:bar>

<m:barPr>

<m:pos m:val="bot"/>

</m:barPr>

<m:e>

<m:r>

<m:t>x</m:t>

</m:r>

</m:e>

</m:bar>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_BarPr) is located in §A.6.1. *End note*]

#### baseJc (Matrix Base Justification)

This element specifies the justification of the matrix. Text outside of the matrix can be aligned with the bottom, top, or center of a matrix object. If this element is omitted, the matrix assumes center justification. In other words, whether the element is absent or present without the val attribute, the default of the val attribute is center.

[*Example*:

This matrix has center baseJc:

This matrix has top baseJc:

This matrix has bottom baseJc:

The XML below represents the matrix with top baseJc:

<m:d>

<m:dPr>

<m:shp m:val="match"/>

</m:dPr>

<m:e>

<m:m>

<m:mPr>

<m:baseJc m:val="top"/>

<m:mcs>

<m:mc>

<m:mcPr>

<m:count m:val="2"/>

<m:mcJc m:val="center"/>

</m:mcPr>

</m:mc>

</m:mcs>

</m:mPr>

<m:mr>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>1</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>2</m:t>

</m:r>

</m:e>

</m:mr>

<m:mr>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>3</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>4</m:t>

</m:r>

</m:e>

</m:mr>

<m:mr>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>5</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>6</m:t>

</m:r>

</m:e>

</m:mr>

</m:m>

</m:e>

</m:d>

Because the matrix base justification is top, the top row of the matrix is lined up with the baseline of the rest of the line.

*End example*]

|  |  |
| --- | --- |
| Attributes | Description |
| val (Value) | Specifies the vertical justification parent element respect to surrounding text. Possible values are top, bottom and center. [*Example*: The following examples illustrate baseJc on the matrix object m.  This matrix has center baseJc:  This matrix has top baseJc:  This matrix has bottom baseJc:  The possible values for this attribute are defined by the ST\_YAlign simple type (§22.9.2.20). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_YAlign) is located in §A.6.1. *End note*]

#### begChr (Delimiter Beginning Character)

This element specifies the beginning, or opening, delimiter character. Mathematical delimiters are enclosing characters such as parentheses, brackets and braces. If this element is omitted, the default begChr is '('. In other words, when the element is absent, the default is for the delimiter object beginning character to be Unicode character U+0028 (LEFT PARENTHESIS).

When the element is present and the val attribute is absent, the default of the val attribute is empty which means the delimiter object beginning character is absent.

[*Example*: In the following example, {𝑎} uses { and } as its enclosing characters:

<m:dPr>

<m:begChr m:val="{"/>

<m:endChr :val="}"/>

</m:dPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the character used by the parent element. When it is omitted, the parent uses its assigned default.  [*Example*: Accent object ã:  <m:accPr>  <m:chr m:val="&#771;"/>  </m:accPr>  *End example*]  The possible values for this attribute are defined by the ST\_Char simple type (§22.1.3.3). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_Char) is located in §A.6.1. *End note*]

#### borderBox (Border-Box Object)

This element specifies the Border Box object, consisting of a border drawn around an instance of mathematical text (such as a formula or equation), as in . If borderBoxPr is omitted, then the default behavior of borderBox is a rectangular border (as shown in the “abc” example below).

[*Example*: The following example shows the XML representation of the following Border Box:

<m:borderBox>

<m:e>

<m:r>

<m:t>abc</m:t>

</m:r>

</m:e>

</m:borderBox>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_BorderBox) is located in §A.6.1. *End note*]

#### borderBoxPr (Border-Box Properties)

This element specifies the properties of the Border Box object, which dictate the types of lines that can be drawn as part of the border. [Example: (Diagonal Strikethrough from Top Left) and (no left or right edges). *End example*]

[*Example*: (left and bottom edges) is represented by the following XML

<m:borderBox>

<m:borderBoxPr>

<m:hideTop m:val="1"/>

<m:hideRight m:val="1"/>

</m:borderBoxPr>

<m:e>

<m:r>

<m:t>x</m:t>

</m:r>

</m:e>

</m:borderBox>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_BorderBoxPr) is located in §A.6.1. *End note*]

#### box (Box Object)

This element specifies the box object, which is used to group components of an equation or other instance of mathematical text. A boxed object can (for example) serve as an operator emulator with or without an alignment point, serve as a line break point, have associated argSz, or be grouped such as not to allow line breaks within. If boxPr is omitted, all properties will be “false” by default.

[*Example*: The mathematical text 𝑎 == 𝑏 uses a box around the double equal sign.

Its XML representation is as follows:

<m:r>

<m:t>a</m:t>

</m:r>

<m:box>

<m:boxPr>

<m:opEmu m:val="1"/>

<m:aln/>

</m:boxPr>

<m:e>

<m:r>

<m:t>==</m:t>

</m:r>

</m:e>

</m:box>

<m:r>

<m:t>b</m:t>

</m:r>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_Box) is located in §A.6.1. *End note*]

#### boxPr (Box Properties)

This element specifies properties of the Box object, for example, whether the Box serves as operator emulator with or without an alignment point, serves as a line break point, or receives the correct spacing for the mathematical differential. [*Example*: is partly represented by the following XML:

<m:box>

<m:boxPr>

<m:opEmu m:val="1"/>

</m:boxPr>

<m:e>

<m:r>

<m:t>x</m:t>

</m:r>

</m:e>

</m:box>

The x is set to be an operator emulator, so it is given operator spacing around it.

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_BoxPr) is located in §A.6.1. *End note*]

#### brk (Break)

This element specifies whether there is a line break at the start of a run, or at the start of the Box object, such that the line wraps at the start of the run or box object. These user-defined line breaks occur when the XML tag <m:brk/> is encountered and does not follow a mathematical "order of precedence". If this element is omitted, a manual break is not inserted. In other words, when the element is absent, the default is for the parent structure to not manually break onto the next line. When the element is present and the val attribute is absent, the default of the val attribute is0meaning that this property’s parent structure manually breaks onto the next line and is aligned with the beginning of the previous line.

The line may happen to wrap at this point if the mathematical text exceeds the column width. OMML does not specify behaviour for any kind of automatic line breaking. Instead, it delegates this functionality to the underlying rendering engine similarly to how it does with the choice of operator spacing and the determination of whether an operator is unary or binary.

[*Example*: The following example includes a manual user-defined line break at the operator emulator:

<m:r>

<m:t>a</m:t>

</m:r>

<m:box>

<m:boxPr>

<m:opEmu m:val="1"/>

<m:brk/>

</m:boxPr>

<m:e>

<m:r>

<m:t>==</m:t>

</m:r>

</m:e>

</m:box>

<m:r>

<m:t>b</m:t>

</m:r>

There is a break on the == symbol so a manual line break occurs right before the == symbol.

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| alnAt (Index of Operator to Align  To) | Specifies the index of the operator on the previous line of mathematical text which shall be used as the alignment point for the current line of mathematical text . A line can be aligned to any operator on the previous line; this attribute specifies exactly which operator shall be the target of that alignment in cases where there are multiple operators. If alnAt is omitted, then all runs (r tag) that follow a brk tag will align with the left margin of the first run of mathematical text.  [*Example*: For example, consider the break in this instance of mathematical text:  𝑎 + 𝑏 + 𝑐 + 𝑑 + 𝑒  + 𝑓 + 𝑔  The second line could theoretically be aligned to any of the four operators in the previous line.  Specifying an alnAt value of 3 for the second line resolves this ambiguity; the second line is aligned to the third operator in the previous line. *End example*]  The possible values for this attribute are defined by the ST\_Integer255 simple type (§22.1.3.6). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_ManualBreak) is located in §A.6.1. *End note*]

#### brkBin (Break on Binary Operators)

This element specifies how binary operators are treated when they coincide with a line break. If this element is omitted, the line break occurs before the binary operator. That is, the binary operator is the first element on the wrapped line.

Whether the element is absent or present without the val attribute, the default of the val attribute is before.

[*Example*: For example:

|  |  |  |
| --- | --- | --- |
| 𝑓 | 𝑓 | 𝑓 |
| 𝑎𝑛𝑛 | 𝑛𝑛 | 𝑎𝑛𝑛 |
| before | after | repeat |

The above examples demonstrate how such an equation would look depending on the value of the brkBin document-property. These actual examples do not rely on the brkBin property of this document to maintain their configurations. This would be impossible because the brkBin property can only be set to one value for an entire document at any given time. Here we are demonstrating all three.

The XML below demonstrates brkBin in use under mathPr:

<m:mathPr>

<m:mathFont m:val="Cambria Math"/>

<m:brkBin m:val="before"/>

<m:brkBinSub m:val="--"/>

<m:smallFrac m:val="0"/>

<m:dispDef/>

<m:lMargin m:val="0"/>

<m:rMargin m:val="0"/>

<m:defJc m:val="centerGroup"/>

<m:wrapIndent m:val="1440"/>

<m:intLim m:val="subSup"/>

<m:naryLim m:val="undOvr"/> </m:mathPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies where to break on binary operators. Possible values are before, after and repeat.  The possible values for this attribute are defined by the ST\_BreakBin simple type (§22.1.3.1). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_BreakBin) is located in §A.6.1. *End note*]

#### brkBinSub (Break on Binary Subtraction)

This element specifies how the subtraction operator is treated when it coincides with a line break, when brkBin is set to repeat. If this element is omitted, the subtraction operator is repeated before and after the break.

Whether the element is absent or present without the val attribute, the default of the val attribute is--.

[*Example*: The XML below demonstrates brkBinSub in use under mathPr:

<m:mathPr>

<m:mathFont m:val="Cambria Math"/>

<m:brkBin m:val="before"/>

<m:brkBinSub m:val="--"/>

<m:smallFrac m:val="0"/>

<m:dispDef/>

<m:lMargin m:val="0"/>

<m:rMargin m:val="0"/>

<m:defJc m:val="centerGroup"/>

<m:wrapIndent m:val="1440"/>

<m:intLim m:val="subSup"/>

<m:naryLim m:val="undOvr"/>

</m:mathPr>

*End example*]

[*Example*:

Given the following XML representation using brkBinSub:

<m:brkBinSub m:val="-+"/>

Because the document option for break on binary subtraction is set to -+, any breaks in Math paragraphs on subtraction operators will display a – on the line before the break and a + on the line after the break. *End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies how the subtraction operator is treated when it coincides with a line break, when brkBin is set to repeat. Possible values are--,-+ and+-.  The possible values for this attribute are defined by the ST\_BreakBinSub simple type (§22.1.3.2). |

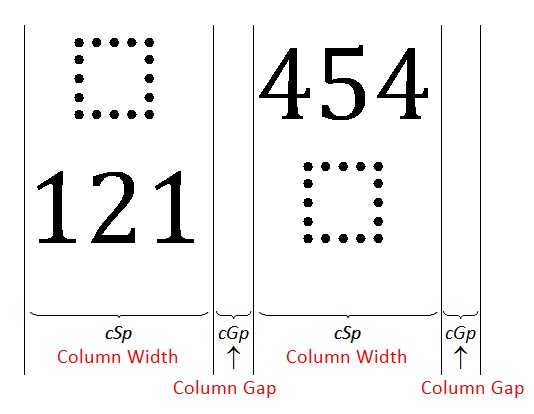
[*Note*: The W3C XML Schema definition of this element’s content model (CT\_BreakBinSub) is located in §A.6.1. *End note*]

#### cGp (Matrix Column Gap)

This element represents the (custom) column gap spacing information; the default value is 0 (which corresponds to 1 em). This value is interpreted differently depending on the value of cGpRule (§22.1.2.19). cGpis not used unless the value ofcGpRule is 3 or 4. When cGpRule is omitted, the default spacing between matrix columns is 1 em (a val attribute value of 0).

Whether the element is absent or present without the val attribute, the default of the val attribute is 0.

The cGpgap spacing (also referred to as “Column Gap” or “Gap Width”) is added to the cSp (Minimum Matrix Column Width) to determine the total Matrix Column Spacing (distance between the same edges of different columns). The following image depicts how cGp and cSpwork together to define matrix column spacing in a 2x2 matrix:



[*Example*: With a cGpRule value of 4, the matrix: has 3 ems of spacing between columns (3 ems = 6\* 0.5 ems). The matrix properties that demonstrate this element in use are:

<m:mPr>

<m:cGpRule m:val="4"/>

<m:cGp m:val="6"/>

<m:mcs>

<m:mc>

<m:mcPr>

<m:count m:val="2"/>

<m:mcJc m:val="center"/>

</m:mcPr>

</m:mc>

</m:mcs> </m:mPr>

*End example*]

See the documentation for cGpRule for more information about how cGp is interpreted for each value of cGpRule.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the amount of space between columns of the parent element (for cGp/cSp) or rows (for rSp). The manner in which this value is determined depends on the setting of the rule of the parent element.  For cGp, if the rule is set to3(or "Exactly"), then the unit is interpreted as twips. If the rule is set to4(or "Multiple"), then the unit is interpreted as number of 0.5 em increments.  For cSp, this value is measured in twips. There is no corresponding cSpRule.  For rSp, if the rule is set to3(or "Exactly"), then the unit is interpreted as points. If the rule is set to4(or "Multiple"), then the unit is interpreted as half-lines.  The possible values for this attribute are defined by the ST\_UnSignedInteger simple type (§22.1.3.14). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_UnSignedInteger) is located in §A.6.1. *End note*]

#### cGpRule (Matrix Column Gap Rule)

This element specifies the type of gap (horizontal spacing) between columns of a matrix; the default is 0.

Horizontal spacing units can be ems or points (stored as twips).

Whether the element is absent or present without the val attribute, the default of the val attribute is 0 (or single-spacing gap).

[*Example*: The following matrix has double spacing (2 ems) between columns:

The XML that represents this property in use is:

<m:mPr>

<m:cGpRule m:val="2"/>

<m:mcs>

<m:mc>

<m:mcPr>

<m:count m:val="2"/>

<m:mcJc m:val="center"/>

</m:mcPr>

</m:mc>

</m:mcs> </m:mPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the type of spacing between rows and/or columns. Possible values are 0, 1, 2, 3, or 4, whose definitions are contained in the following table:   |  |  |  | | --- | --- | --- | | **Value** | **Column/Row Gap** | **Example** | | 0 | Single spacing gap (1 em) | 1 2 | | 1 | 1.5 spacing gap (1.5 ems) | 1 2 | | 2 | 2 spacing gap (2 ems) | 1 2 | | 3 | Exactly  (for columns, rely on value of cGp, measured in twips) (for rows, rely on value of rSp, measured in points) | 1 2 | | 4 | Multiple  (for columns, rely on value of cGp, measured in  0.5 em increments) (for rows, rely on value of rSp, measured in lines) | 1 2 |   The possible values for this attribute are defined by the ST\_SpacingRule simple type (§22.1.3.11). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_SpacingRule) is located in §A.6.1. *End note*]

#### chr (Character)

This element specifies the character to be attached to the base of an accent object, a group character object, or an n-ary operator object. When the parent element is accPr, the chr value should be within the range of (U+0300–U+036F) or (U+20D0–U+20EF). When the parent element is group ChrPr, the chr value should be a horizontal stretch character, such as U+2190 (LEFTWARD ARROW). When the parent element is naryPr, the chr value should be an n-ary operator such as U+222B (INTEGRAL).

When the element is present and the val attribute is absent, the default of the val attribute is empty which means the corresponding objects character is absent.

If this property is omitted for accPr, the default accent character is U+0302 (ACCENT, COMBINING CIRCUMFLEX). If this property is omitted for group ChrPr, the default character is U+23DF (BOTTOM CURLY BRACKET). If this property is omitted for naryPr, the default character is U+222B (INTEGRAL).

In other words, when the element is absent, the defaults are as follows:

* accent object – Unicode character U+0302 (ACCENT, COMBINING CIRCUMFLEX)
* group-character object – Unicode character U+23DF (BOTTOM CURLY BRACKET)
* n-ary operator object – Unicode character U+222B (INTEGRAL)

[*Example*: Examples of accent characters are the dot, hat and arrow in the following cases:𝑎̇ 𝑎̂ 𝑎⃗.

For example, the following XML represents the acc 𝑎̃.

<m:acc>

<m:accPr>

<m:chr m:val="&#771;"/>

</m:accPr>

<m:e>

<m:r>

<m:t>a</m:t>

</m:r>

</m:e>

</m:acc>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the character used by the parent element. When it is omitted, the parent uses its assigned default.  [*Example*: Accent object ã:  <m:accPr>  <m:chr m:val="&#771;"/>  </m:accPr>  *End example*]  The possible values for this attribute are defined by the ST\_Char simple type (§22.1.3.3). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_Char) is located in §A.6.1. *End note*]

#### count (Matrix Column Count)

This element specifies the number of columns to which a property applies.

Whether the element is absent or present without the val attribute, the default of the val attribute is1.

[*Example*: The example below specifies that two of the columns in the matrix described by the XML have the center property.

<m:mPr>

<m:cSp m:val="120"/>

<m:mcs>

<m:mc>

<m:mcPr>

<m:count m:val="2"/>

<m:mcJc m:val="center"/>

</m:mcPr>

</m:mc>

</m:mcs> </m:mPr>

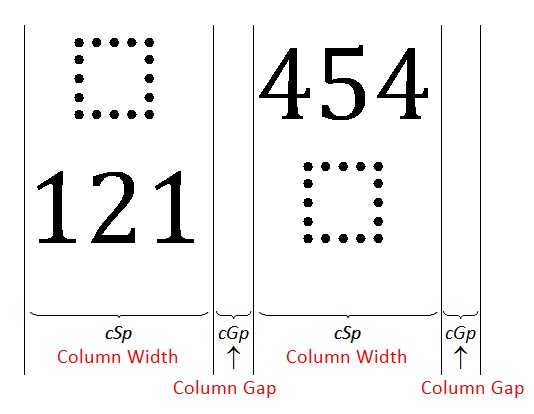
*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the number of columns to which a column property applies.  [*Example*: A count attribute value of 3 specifies that the property applies to the next three columns of the matrix. *End example*]  The possible values for this attribute are defined by theST\_Integer255simple type (§22.1.3.6). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_Integer255) is located in §A.6.1. *End note*]

#### cSp (Minimum Matrix Column Width)

This element specifies the minimum column width of a matrix. The actual column width of a matrix will be the greater of either the width of the column’s widest argument or cSp. This additional spacing can be added to enhance appearance. If this element is omitted, the default minimum column width is 0. Whether the element is absent or present without the val attribute, the default of the val attribute is 0. The cGp gap spacing (also referred to as “Column Gap” or “Gap Width”) is added to the cSp(Minimum Matrix Column Width) to determine the total Matrix Column Spacing (distance between the same edges of different columns). The value of cSp is interpreted as twips (a twip is 1/20th of a point). Therefore, a spacing of1point will be set by a cSp value of 20. This is the only use for cSp. There is no corresponding cSp Rule. The following image depicts how cGp and cSp work together to define matrix column spacing in a 2x2 matrix:



[*Example*: The following XML specifies that there should never be fewer than 6 pts (120 twips) between adjacent column edges of the matrix:

<m:mPr>

<m:cSp m:val="120"/>

<m:mcs>

<m:mc>

<m:mcPr>

<m:count m:val="2"/>

<m:mcJc m:val="center"/>

</m:mcPr>

</m:mc>

</m:mcs> </m:mPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the amount of space between columns of the parent element (for cGp/cSp) or rows (for rSp). The manner in which this value is determined depends on the setting of the rule of the parent element.  For cGp, if the rule is set to 3 (or "Exactly"), then the unit is interpreted as twips. If the rule is set to 4 (or "Multiple"), then the unit is interpreted as number of 0.5 em increments.  For cSp, this value is measured in twips. There is no corresponding cSpRule.  For rSp, if the rule is set to 3 (or "Exactly"), then the unit is interpreted as points. If the rule is set to 4 (or "Multiple"), then the unit is interpreted as half-lines.  The possible values for this attribute are defined by the ST\_UnSignedInteger simple type (§22.1.3.14). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_UnSignedInteger) is located in §A.6.1. *End note*]

#### ctrlPr (Control Properties)

This element specifies properties on control characters; that is, object characters that cannot be selected. Examples of control characters are n-ary operators (excluding their limits and bases), fraction bars (excluding the numerator and denominator) and grouping characters (excluding the base). ctrlPr allows formatting properties to be stored on these control characters. The control character inherits its formatting from the paragraph formatting; ctrlPr contains the formatting differences between the control character and the paragraph formatting.

If this element is omitted, the character properties of the first control character are the same as the first character in the math object.

[*Example*: The example below shows that the control character is of font Cambria Math. All other formatting, such as text size and color, are the same as the paragraph.

<m:ctrlPr>

<w:rPr>

<w:rFonts w:ascii="Cambria Math" w:hAnsi="Cambria Math"/>

</w:rPr>

</m:ctrlPr>

*End example*]

CtrlPr is also used to save properties on characters used in the built-down form of an instance of mathematical text that are not displayed in Professional form. For example, the mathematical text (in linear form) ∫\_0^1 might have color on the \_ or ^. Though these characters are not displayed in Professional form, their formatting is stored such that their formatting will roundtrip through build up and build down.

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_CtrlPr) is located in §A.6.1. *End note*]

#### d (Delimiter Object)

This element specifies the delimiter object, consisting of opening and closing delimiters (such as parentheses, braces, brackets and vertical bars) and an element contained inside. The delimiter may have more than one element, with a designated separator character between each element. [*Example*:

Delimiter with one base:

Delimiter with more than one base and separators, whose XML is shown below:

<m:d>

<m:e>

<m:sSup>

<m:e>

<m:r>

<m:t>x</m:t>

</m:r>

</m:e>

<m:sup>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>2</m:t>

</m:r>

</m:sup>

</m:sSup>

</m:e>

<m:e>

<m:sSup>

<m:e>

<m:r>

<m:t>y</m:t>

</m:r>

</m:e>

<m:sup>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>2</m:t>

</m:r>

</m:sup>

</m:sSup>

</m:e>

</m:d>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_D) is located in §A.6.1. *End note*]

#### defJc (Default Justification)

This element specifies the default justification of display math, at the document level. Individual instances of mathematical text can overrule the default setting. If this element is omitted, the default justification is centerGroup. Whether the element is absent or present without the val attribute, the default of the val attribute is centerGroup.

Display math can be left justified, right justified, centered, or centered as a group. When display math is centered as a group, the mathematical text is left aligned within a block and the entire block is centered with respect to column margins. If this element is omitted, the mathematical text is centered as a group. [*Example*: The XML below demonstrates defJc in use:

<m:mathPr>

<m:mathFont m:val="Cambria Math"/>

<m:brkBin m:val="before"/>

<m:brkBinSub m:val="--"/>

<m:smallFrac m:val="0"/>

<m:dispDef/>

<m:lMargin m:val="0"/>

<m:rMargin m:val="0"/>

<m:defJc m:val="centerGroup"/>

<m:wrapIndent m:val="1440"/>

<m:intLim m:val="subSup"/>

<m:naryLim m:val="undOvr"/>

</m:mathPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the default justification of mathematical text in the document. Possible values are center, centerGroup, left and right.  The possible values for this attribute are defined by the ST\_Jc simple type (§22.1.3.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OMathJc) is located in §A.6.1. *End note*]

#### deg (Degree)

This element specifies the degree in the mathematical radical. This element is optional. When omitted, the square root function, as in, is assumed. [*Example*:

The 3 in is represented by the XML below:

<m:rad>

<m:deg>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>3</m:t>

</m:r>

</m:deg>

<m:e>

<m:r>

<m:t>x</m:t>

</m:r>

</m:e>

</m:rad>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OMathArg) is located in §A.6.1. *End note*]

#### degHide (Hide Degree)

This element specifies the per-object option to hide the degree of a radical. Every rad has a deg, but the deg can appear or not appear. When degHide is set to 1 or true, the degree is not shown, as in (XML shown below). When degHide is omitted, the default is 0 or false; that is, the degree is not hidden. In other words, when the element is absent, the default value of the property is 0 meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied.

[*Example*:

<m:rad>

<m:radPr>

<m:degHide m:val="1"/>

</m:radPr>

<m:deg>

</m:deg>

<m:e>

<m:r>

<m:t>x</m:t>

</m:r>

</m:e>

</m:rad>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### den (Denominator)

This element specifies the denominator of a fraction. [*Example*: For example, the 𝑏 in :

<m:f>

<m:fPr>

<m:type m:val="skw"/>

</m:fPr>

<m:num>

<m:r>

<m:t>a</m:t>

</m:r>

</m:num>

<m:den>

<m:r>

<m:t>b</m:t>

</m:r>

</m:den>

</m:f>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OMathArg) is located in §A.6.1. *End note*]

#### diff (Differential)

The element specifies the differential property on box. When 1 or true, the box acts as a differential (e.g., 𝑑𝑥 in an integrand) and receives the appropriate horizontal spacing for the mathematical differential. When this property is omitted, the box is not treated as a differential.

When the element is absent, the default value of the property is 0 meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied.

[*Example*: The following example demonstrates a box set as differential in use, both in its proper form and in XML:

<m:nary>

<m:naryPr>

<m:chr m:val="&#8747;"/>

</m:naryPr>

<m:sub>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>0</m:t>

</m:r>

</m:sub>

<m:sup>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>1</m:t>

</m:r>

</m:sup>

<m:e>

<m:r>

<m:t>x</m:t>

</m:r>

<m:box>

<m:boxPr>

<m:diff m:val="1"/>

</m:boxPr>

<m:e>

<m:r>

<m:t>dx</m:t>

</m:r>

</m:e>

</m:box>

</m:e>

</m:nary>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### dispDef (Use Display Math Defaults)

This element specifies the document-level property to overwrite paragraph settings for mathematical text. When omitted, this element is set to1ortrueand special math settings are applied. Whether the element is absent or present without the val attribute, the default of the val attribute is1meaning that this option is applied.

[*Example*: The XML below demonstrates dispDef in use:

<m:mathPr>

<m:mathFont m:val="Cambria Math"/>

<m:brkBin m:val="before"/>

<m:brkBinSub m:val="--"/>

<m:smallFrac m:val="0"/>

<m:dispDef/>

<m:lMargin m:val="0"/>

<m:rMargin m:val="0"/>

<m:defJc m:val="centerGroup"/>

<m:wrapIndent m:val="1440"/>

<m:intLim m:val="subSup"/>

<m:naryLim m:val="undOvr"/> </m:mathPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### dPr (Delimiter Properties)

This element specifies the properties of d, including the enclosing and separating characters and the properties that affect the shape of the delimiters.

[*Example*: is represented by the following XML:

<m:d>

<m:dPr>

<m:begChr m:val="["/>

<m:endChr m:val="]"/>

</m:dPr>

<m:e>

<m:r>

<m:t>a+b</m:t>

</m:r>

</m:e>

</m:d>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_DPr) is located in §A.6.1. *End note*]

#### e (Element (Argument))

This tag, which is an abbreviation for “element”, serves several functions (18 total) including that of the base argument of a mathematical object or function, the elements in an array and the elements in boxes. If all subelements are omitted, this element specifies the presence of an empty argument.

[*Example*: The numerator of this function is an empty argument: . *End example*] [*Note*: The tag was shortened to e to improve readability over “element”. “Element” was preferred over “Base” because “Element” sounded more neutral and appropriate to its multiple uses. *End note*]

The following table shows the 18 different uses for e:

|  |  |
| --- | --- |
| **Parent Element** | **Use** |
| acc | Accent base argument |
| bar | Argument to which the bar is applied |
| borderBox | Argument around which the border box is drawn |
| box | Argument inside the abstract box |
| d | Argument inside the delimiters |
| eqArr | Each instance of mathematical text in the single-column array |
| func | Math argument list of the function |
| groupChr | Group character base |
| limLow | Base of the lower limit |
| limUpp | Base of the upper limit |
| mr | Each element in the matrix row |
| nary | n-ary and, e.g., integrand for an integral, summand for a summation |
| phant | Argument for the phantom |
| rad | Radicand |
| sPre | Base of the prescript object |
| sSub | Base of the subscript object |
| sSubSup | Base of the subsup object |
| sSup | Base of the superscript object |

[*Example*: For example, the func has fName sin and e x:

<m:func>

<m:fName>

<m:r>

<m:t>sin</m:t>

</m:r>

</m:fName>

<m:e>

<m:r>

<m:t>x</m:t>

</m:r>

</m:e>

</m:func>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OMathArg) is located in §A.6.1. *End note*]

#### endChr (Delimiter Ending Character)

This element specifies the ending, or closing, delimiter character. Mathematical delimiters are enclosing characters such as parentheses, brackets and braces. If this element is omitted, the default endChr is ')'. In other words, when the element is absent, the default is for the delimiter object beginning character to be Unicode character U+0029 (RIGHT PARENTHESIS).

When the element is present and the val attribute is absent, the default of the val attribute is empty which means the delimiter object beginning character is absent.

[*Example*: In the following example, uses { and } as its enclosing characters:

<m:dPr>

<m:begChr m:val="{"/>

<m:endChr m:val="}"/> </m:dPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the character used by the parent element. When it is omitted, the parent uses its assigned default.  [*Example*: Accent object ã:  <m:accPr>  <m:chr m:val="&#771;"/>  </m:accPr>  *End example*]  The possible values for this attribute are defined by the ST\_Char simple type (§22.1.3.3). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_Char) is located in §A.6.1. *End note*]

#### eqArr (Array Object)

This element specifies the Array object (sometimes referred to as "Equation Array", despite its ability to hold mathematical text other than equations), an object consisting of one or more equations, expressions, or other mathematical text runs that can be vertically justified as a unit with respect to surrounding text on the line. Alignment of multiple points within each run of mathematical text can occur within the array through the use of align values and spacer values. An *align value* is an ampersand within the array which acts as an alignment point (as described in §22.1.2.3). A *spacer value* is an ampersand (represented by "&amp;" in the example below) within the array which designates where space can be added in order to align the align values on different rows of the array. Within each argument in the array, every odd ampersand is an align value and every even ampersand is a spacer value (as well, the beginning of each argument provides an implied spacer value). If eqArrPr is omitted, then the default values for its properties will be used. These defaults are:

|  |  |
| --- | --- |
| **Property** | **Default Value** |
| baseJc | “center” |
| ctrlPr | <The character property of the first control character will be the character property of the first character in the eqArr object > |
| maxDist | “0” |
| objDist | “0” |
| rSp | “0” |
| rSpRule | “0” <single> |

[*Example*:

A simple array, is represented as follows:

<m:eqArr>

<m:e>

<m:r>

<m:t>a=b+c</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:t>d+e=f</m:t>

</m:r>

</m:e>

</m:eqArr>

A more complex array with alignment points, e.g.:

is represented as follows:

<m:eqArr>

<m:e>

<m:r>

<m:t>&amp;x+&amp;&amp;y+&amp;&amp;z&amp;&amp;= &amp;&amp;10</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:t>3&amp;x+&amp;&amp;y+&amp;2&amp;z&amp;&amp;= &amp;&amp;34</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:t>-5&amp;x+&amp;2&amp;y-&amp;&amp;z&amp;&amp;= &amp;&amp;-

14</m:t>

</m:r>

</m:e>

</m:eqArr>

Notice that the variables, operators and digits of the sums line up properly, as the align values line up between each argument by adding space at the position of each spacer value.

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_EqArr) is located in §A.6.1. *End note*]

#### eqArrPr (Array Properties)

This element specifies the properties of the array object, including the vertical justification of the object and layout inside the object.

[*Example*:

<m:eqArr>

<m:eqArrPr>

<m:baseJc m:val="bottom"/>

</m:eqArrPr>

<m:e>

<m:r>

<m:t>111&amp;a&amp;&amp;&amp;&amp;&amp;&amp;=&amp;&amp;c</m:t>

</m:r>

</m:e>

<m:e>

<m:r> <m:t>111&amp;a&amp;&amp;+&amp;111&amp;b&amp;&amp;=&amp;111&amp;c</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:t>&amp;&amp;&amp;&amp;&amp;b&amp;&amp;=&amp;111&amp;c</m:t>

</m:r> </m:e>

</m:eqArr>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_EqArrPr) is located in §A.6.1. *End note*]

#### f (Fraction Object)

This element specifies the fraction object, consisting of a numerator and denominator separated by a fraction bar. The fraction bar can be horizontal or diagonal, depending on the fraction properties. The fraction object is also used to represent the stack function, which places one element above another, with no fraction bar.

[*Example*: Examples of fractions are:

Bar Fraction:

Skewed Fraction:

Linear Fraction:

No-Bar Fraction (Stack):

The fractionis represented as:

<m:f>

<m:fPr>

<m:type m:val="skw"/>

</m:fPr>

<m:num>

<m:r>

<m:t>a</m:t>

</m:r>

</m:num>

<m:den>

<m:r>

<m:t>b</m:t>

</m:r>

</m:den>

</m:f>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_F) is located in §A.6.1. *End note*]

#### fName (Function Name)

This element specifies the name of the function in the Function-Apply object func. For example, function names are sin and cos.

[*Example*: For example, the func has fName sin and e x:

<m:func>

<m:fName>

<m:r>

<m:t>sin</m:t>

</m:r>

</m:fName>

<m:e>

<m:r>

<m:t>x</m:t>

</m:r>

</m:e>

</m:func>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OMathArg) is located in §A.6.1. *End note*]

#### fPr (Fraction Properties)

This element specifies the properties of the fraction object f. Properties of the Fraction object include the type or style of the fraction. The fraction bar can be horizontal or diagonal, depending on the fraction properties. The fraction object is also used to represent the stack function, which places one element above another, with no fraction bar. [*Example*: Examples of fractions are:

[*Example*: Examples of fractions are:

Bar Fraction:

Skewed Fraction:

Linear Fraction:

No-Bar Fraction (Stack):

The fractionis represented as:

<m:f>

<m:fPr>

<m:type m:val="skw"/>

</m:fPr>

<m:num>

<m:r>

<m:t>a</m:t>

</m:r>

</m:num>

<m:den>

<m:r>

<m:t>b</m:t>

</m:r>

</m:den>

</m:f>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_FPr) is located in §A.6.1. *End note*]

#### func (Function Apply Object)

This element specifies the Function-Apply object, which consists of a function name and an argument element (e) acted upon. It is often applied using a form of linear format. For example, in the linear format described in Unicode Technical Article #28, this object is applied by using the Function Application character (U+2061). [*Example*: Examples of Function-Apply objects include: , and .

[*Example*: For example, the func has fName sin and e x:

<m:func>

<m:fName>

<m:r>

<m:t>sin</m:t>

</m:r>

</m:fName>

<m:e>

<m:r>

<m:t>x</m:t>

</m:r>

</m:e>

</m:func>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_Func) is located in §A.6.1. *End note*]

#### funcPr (Function Properties)

This element specifies properties such as ctrlPr that can be stored on the function apply object func.

[*Example*: is represented by the following XML:

<m:func>

<m:funcPr/>

<m:fName>

<m:r>

<m:rPr>

<m:sty m:val="p"/>

</m:rPr>

<m:t>tan</m:t>

</m:r>

</m:fName>

<m:e>

<m:r>

<m:t>x</m:t>

</m:r>

</m:e>

</m:func>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_FuncPr) is located in §A.6.1. *End note*]

#### groupChr (Group-Character Object)

This element specifies the Group-Character object, consisting of a character drawn above or below text, often with the purpose of visually grouping items.

[*Example*: The following example demonstrates the groupChr in use, both in its proper form and in XML:

<m:groupChr>

<m:groupChrPr>

<m:chr m:val="&#9182;"/>

<m:pos m:val="top"/>

</m:groupChrPr>

<m:e>

<m:r>

<m:t>x+x+&#22EF;</m:t>

</m:r>

</m:e>

</m:groupChr>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_GroupChr) is located in §A.6.1. *End note*]

#### groupChrPr (Group-Character Properties)

This element specifies the properties of the Group-Character object groupChr. These properties can be used to specify the character placed above or below the argument and the position of the character. When omitted, character⏟(U+23DF, BOTTOM CURLY BRACKET) is used as the chr and its pos is set to bot.

[*Example*:

is represented by the following XML:

<m:groupChr>

<m:groupChrPr>

<m:chr m:val="⏞"/>

<m:pos m:val="top"/>

<m:vertJc m:val="bot"/>

</m:groupChrPr>

<m:e>

<m:r>

<m:t>x+y</m:t>

</m:r>

</m:e>

</m:groupChr>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_GroupChrPr) is located in §A.6.1. *End note*]

#### grow (n-ary Grow)

This element specifies the growth property of n-ary operators. When 0 or false, n-ary operators such as integrals and summations do not grow to match the size of their operand height. When 1 or true, the n-ary operator grows vertically to match its operand height. If this property is omitted, grow is set to 0.

When the element is absent, the default value of the property when it is a child of a delimiter object is1(meaning that this property is applied) and when it is a child of an n-ary operator object it is0(meaning that this property is not applied).

When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this property is applied.

[*Example*: The two integrals below demonstrate the difference between grow=0 and grow=1.

The XML that defines nary growth is:

<m:naryPr>

<m:chr m:val="&#8747;"/>

<m:grow m:val="1"/> </m:naryPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### hideBot (Hide Bottom Edge)

This element specifies the hidden or shown state of the bottom edge of borderBox. When this element is omitted, the bottom edge is shown. In other words, when the element is absent, the default value of the property is 0 meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this property is applied.

When applied, the bottom border is hidden, as in .

[Example:

<m:borderBox>

<m:borderBoxPr>

<m:hideBot/>

</m:borderBoxPr>

<m:e>

<m:r>

<m:t>abc</m:t>

</m:r>

</m:e>

</m:borderBox>

*End example*]

Shared MLs Reference Material

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### hideLeft (Hide Left Edge)

This element specifies the hidden or shown state of the left edge of borderBox. When this element is omitted, the edge is shown. In other words, when the element is absent, the default value of the property is0meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this property is applied.

When applied, the left border is hidden, as in . [*Example*:

<m:borderBox>

<m:borderBoxPr>

<m:hideLeft/>

</m:borderBoxPr>

<m:e>

<m:r>

<m:t>abc</m:t>

</m:r>

</m:e>

</m:borderBox>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### hideRight (Hide Right Edge)

This element specifies the hidden or shown state of the right edge of borderBox. When this element is omitted, the edge is shown. In other words, when the element is absent, the default value of the property is0meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this property is applied.

When applied, the right border is hidden, as in .

[*Example*:

<m:borderBox>

<m:borderBoxPr>

<m:hideRight/>

</m:borderBoxPr>

<m:e>

<m:r>

<m:t>abc</m:t>

</m:r>

</m:e>

</m:borderBox>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### hideTop (Hide Top Edge)

This element specifies the hidden or shown state of the top edge of borderBox. When this element is omitted, the edge is shown. In other words, when the element is absent, the default value of the property is0meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this property is applied.

When applied, the top border is hidden, as in . [*Example*:

<m:borderBox>

<m:borderBoxPr>

<m:hideTop/>

</m:borderBoxPr>

<m:e>

<m:r>

<m:t>abc</m:t>

</m:r>

</m:e>

</m:borderBox>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### interSp (Inter-Equation Spacing)

This element specifies spacing between equations, expressions, or other instances of mathematical text within a display math paragraph, in twips.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the value, in twips, of the parent element.  The possible values for this attribute are defined by the ST\_TwipsMeasure simple type (§22.9.2.14). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_TwipsMeasure) is located in §A.6.1. *End note*]

#### intLim (Integral Limit Locations)

This element specifies the document setting for the default placement of integral limits, when converted from a linear form to a two-dimensional output (professional form). Limits can be either centered above and below the integral, or positioned just to the right of the operator, as in:

When this integral object is written linearly, as∫ \_𝑎^𝑏, the placement of limits is ambiguous. intLim is a document-level property that specifies the default positioning. When this element is omitted, the default placement of integral limits is subSup (that is, the location of subscripts and superscripts, or just to the right of the base or operator). Whether the element is absent or present without the val attribute, the default of the val attribute is subSup.

[*Example*: The XML that specifies this property in use is:

<m:mathPr>

<m:mathFont m:val="Cambria Math"/>

<m:brkBin m:val="before"/>

<m:brkBinSub m:val="--"/>

<m:smallFrac m:val="0"/>

<m:dispDef/>

<m:lMargin m:val="0"/>

<m:rMargin m:val="0"/>

<m:defJc m:val="centerGroup"/>

<m:wrapIndent m:val="1440"/>

<m:intLim m:val="subSup"/>

<m:naryLim m:val="undOvr"/>

</m:mathPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the default location of limits on the parent object. Possible values are subSup and undOvr.  The possible values for this attribute are defined by the ST\_LimLoc simple type (§22.1.3.8). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_LimLoc) is located in §A.6.1. *End note*]

#### intraSp (Intra-Equation Spacing)

This element specifies the spacing between adjacent display math paragraphs, in twips. If this element is omitted, no spacing is applied between adjacent math paragraphs.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the value, in twips, of the parent element.  The possible values for this attribute are defined by the ST\_TwipsMeasure simple type (§22.9.2.14). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_TwipsMeasure) is located in §A.6.1. *End note*]

#### jc (Justification)

This element specifies justification of the math paragraph (a series of adjacent instances of mathematical text within the same paragraph). A math paragraph can be Left Justified, Right Justified, Centered, or Centered as Group. If this element is omitted, the math paragraph is Centered as Group. Whether the element is absent or present without the val attribute, the default of the val attribute is centerGroup. This means that the instances of mathematical text can be aligned with respect to each other, but the entire group of mathematical text is centered as a whole. [*Example*: An example of Centered as Group is the following example, in which each equation is left-aligned, but the series is centered:

𝑥 = 𝑥1+ 𝑥2+ 𝑥3+ ⋯

𝑦 = 𝑦1+ 𝑦2+ 𝑦3+ 𝑦4+ ⋯ 𝑧 = 𝑧1+ 𝑧2+ 𝑧3+ 𝑧4+ 𝑧5+ ⋯

The XML that demonstrates jc in use is:

<m:oMathParaPr>

<m:jc m:val="centerGroup"/>

</m:oMathParaPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the default justification of mathematical text in the document. Possible values are center, centerGroup, left and right.  The possible values for this attribute are defined by the ST\_Jc simple type (§22.1.3.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OMathJc) is located in §A.6.1. *End note*]

#### lim (Limit)

This element specifies the lower limit of the limLow object and the upper limit of the limUpp function.

[*Example*: For example, the limit of the limLow is 𝑛 → ∞. The XML that specifies this object is:

<m:limLow>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>lim</m:t>

</m:r>

</m:e>

<m:lim>

<m:r>

<m:t>n&#8594;&#8734;</m:t>

</m:r>

</m:lim>

</m:limLow>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OMathArg) is located in §A.6.1. *End note*]

#### limLoc (n-ary Limit Location)

This element specifies the location of limits in n-ary operators. Limits can be either centered above and below the n-ary operator (shown in the first summation below) or positioned just to the right of the operator (shown in the second summation below).

When the element is absent, the default value of this property is ignored and the intLim (for integral n-ary operator objects) and naryLim (for all other n-ary operator objects) options are looked to for the limit placement.

Ultimately, when this element is omitted, the default location is undOvr in display mode and subSup otherwise.

When the element is present and the val attribute is absent, the default of the val attribute undOvr.

[*Example*: The XML representing the relevant properties of the second summation above (subSup) is:

<m:naryPr>

<m:chr m:val="&#8721;"/>

<m:limLoc m:val="subSup"/>

<m:grow m:val="1"/>

</m:naryPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the default location of limits on the parent object. Possible values are subSup and undOvr.  The possible values for this attribute are defined by the ST\_LimLoc simple type (§22.1.3.8). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_LimLoc) is located in §A.6.1. *End note*]

#### limLow (Lower-Limit Object)

This element specifies the Lower-Limit object, consisting of text on the baseline and reduced-size text immediately below it. Examples of limLow include and .

[*Example*: The XML that represents is:

<m:limLow>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>lim</m:t>

</m:r>

</m:e>

<m:lim>

<m:r>

<m:t>n&#8594;&#8734;</m:t>

</m:r>

</m:lim>

</m:limLow>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_LimLow) is located in §A.6.1. *End note*]

#### limLowPr (Lower-Limit Properties)

This element specifies control properties (ctrlPr) that can be stored on the Lower Limit (limLow).

[*Example*: is represented by the following XML:

<m:limLow>

<m:limLowPr/>

<m:e>

<m:r>

<m:t>lim</m:t>

</m:r>

</m:e>

<m:lim>

<m:r>

<m:t>a→∞</m:t>

</m:r>

</m:lim>

</m:limLow>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_LimLowPr) is located in §A.6.1. *End note*]

#### limUpp (Upper-Limit Object)

This element specifies the Upper-Limit object, consisting of text on the baseline and reduced-size text immediately above it.

[*Example*: Examples of limUpp include and.

The XML that specifies the limUpp is:

<m:limUpp>

<m:e>

<m:r>

<m:t>=</m:t>

</m:r>

</m:e>

<m:lim>

<m:r>

<m:rPr>

<m:nor/>

</m:rPr>

<m:t>def</m:t>

</m:r>

</m:lim>

</m:limUpp>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_LimUpp) is located in §A.6.1. *End note*]

#### limUppPr (Upper-Limit Properties)

This element specifies control properties (ctrlPr) that can be stored on the Upper Limit (limUpp).

[*Example*: is represented by the following XML:

<m:limUpp>

<m:limUppPr/>

<m:e>

<m:r>

<m:t>lim</m:t>

</m:r>

</m:e>

<m:lim>

<m:r>

<m:t>a→∞</m:t>

</m:r>

</m:lim>

</m:limUpp>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_LimUppPr) is located in §A.6.1. *End note*]

#### lit (Literal)

This element specifies that the characters in the run are literal; that is, they are to be interpreted literally and not be built up based on any implied mathematical meaning. This is especially useful for operators or other special characters that signal a need for build up to an OMML reader. These characters are often encountered during a given instance of mathematical text when presented in a1-dimensional linear format, such as the linear format defined by Unicode Technical Note #28 (Sargent 2006).

When the element is absent, the default value of the property is0meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this property is applied.

[*Example*: In the following XML, the + operator is treated literally and does not receive proper binary spacing:

<m:r>a</m:r>

<m:r>

<m:rPr>

<m:lit/>

</m:rPr>

<m:t>+</m:t>

</m:r>

<m:r>

<m:t>b</m:t>

</m:r>

As another example, consider the following Office Open XML Math markup:

<m:r>

<m:t>a</m:t>

</m:r>

<m:r>

<m:rPr>

<m:lit/>

</m:rPr>

<m:t>/</m:t>

</m:r>

<m:r>

<m:t>b</m:t>

</m:r>

This would be displayed as: 𝑎/𝑏. If built up, it would normally be converted to ; however, the presence of this 𝑏 element specifies that the solidus should be treated literally, resulting in 𝑎/𝑏. *End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### lMargin (Left Margin)

This element specifies the left margin for math, in twips. If this element is omitted, no left margin is used. In other words, when the element is absent, the default value of the option is 0.

When the element is present and the val attribute is absent, the default of the val attribute is 1440 (or 1 inch).

Math margins are added to the paragraph settings for margins. If the sum of lMargin and rMargin exceed the width available, lMargin should be ignored.

[*Example*: The following XML demonstrates an lMargin setting of 1".

<m:mathPr>

<m:mathFont m:val="Cambria Math"/>

<m:brkBin m:val="before"/>

<m:brkBinSub m:val="--"/>

<m:smallFrac m:val="0"/>

<m:dispDef/>

<m:lMargin m:val="1440"/>

<m:rMargin m:val="0"/>

<m:defJc m:val="centerGroup"/>

<m:wrapIndent m:val="1440"/>

<m:intLim m:val="subSup"/>

<m:naryLim m:val="undOvr"/>

</m:mathPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the value, in twips, of the parent element.  The possible values for this attribute are defined by the ST\_TwipsMeasure simple type (§22.9.2.14). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_TwipsMeasure) is located in §A.6.1. *End note*]

#### m (Matrix Object)

This element specifies the Matrix object, consisting of one or more elements laid out in one or more rows and one or more columns. It is important to note that matrices do not have built in delimiters. Like other math elements, matrices are contained in a delimiter object (§22.1.2.24) when delimiters are desired. Empty arguments (see §22.1.2.32) can be used to create gaps in matrices. The plcHide tag (§22.1.2.83) can be used to indicate whether the empty arguments should be visible in the matrix (see the plcHide documentation for more information). If mPr is omitted, the values of baseJc, cGp, cGpRule, cSp, ctrlPr, mcs, plcHide, rSp and rSpRule are shown in the following table:

|  |  |
| --- | --- |
| **Property** | **Default Value** |
| baseJc | “center” |
| cGp | “0” |
| cGpRule | “0” <single> |
| cSp | “0” |
| ctrlPr | <The character property of the first control character shall be the character property of the first character in the m object > |
| mcs | <All columns will be vertically center aligned> |
| plcHide | “0” |
| rSp | “0” |
| rSpRule | “0” <single> |

[*Example*: Examples of matrices are: and . Below is an example of a 2x2 matrix and its XML in its proper form (surrounding delimiters are not a property of the matrix and are only shown to demonstrate their relationship with a matrix. See the documentation for the d tag for XML examples of delimiters).

<m:d>

<m:m>

<m:mPr>

<m:mcs>

<m:mc>

<m:mcPr>

<m:count m:val="2"/>

<m:mcJc m:val="center"/>

</m:mcPr>

</m:mc>

</m:mcs>

</m:mPr>

<m:mr>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>1</m:t>

</m:r >

</m:e>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>2</m:t>

</m:r >

</m:e>

</m:mr>

<m:mr>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>3</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>4</m:t>

</m:r >

</m:e>

</m:mr>

</m:m>

</m:d>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_M) is located in §A.6.1. *End note*]

#### mathFont (Math Font)

This element specifies the default math font to be used in the document. If this element is omitted, font substitution (§17.8.2) should be used to determine the most appropriate font for use throughout the document.

[*Example*: The XML containing this property is:

<m:mathPr>

<m:mathFont m:val="Cambria Math"/>

<m:brkBin m:val="before"/>

<m:brkBinSub m:val="--"/>

<m:smallFrac m:val="0"/>

<m:dispDef/>

<m:lMargin m:val="1440"/>

<m:rMargin m:val="0"/>

<m:defJc m:val="centerGroup"/>

<m:wrapIndent m:val="1440"/>

<m:intLim m:val="subSup"/>

<m:naryLim m:val="undOvr"/> </m:mathPr>

*End example*]

[Example: Given the following equation: 𝑎 + 𝑏 = 𝑐 and the following XML:

<m:mathFont m:val="Cambria Math"/>

Because the document option for Math font is set to the mathematically enabled font Cambria Math, Math is formatted with that font. *End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the default math font to be used in the document.  The possible values for this attribute are defined by the ST\_String simple type (§22.9.2.13). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_String) is located in §A.6.1. *End note*]

#### mathPr (Math Properties)

This element specifies the document-level properties for all math in the document.

[Example:

<m:mathPr>

<m:mathFont m:val="Cambria Math"/>

<m:brkBin m:val="before"/>

<m:brkBinSub m:val="--"/>

<m:smallFrac m:val="0"/>

<m:dispDef/>

<m:lMargin m:val="0"/>

<m:rMargin m:val="0"/>

<m:defJc m:val="centerGroup"/>

<m:wrapIndent m:val="1440"/>

<m:intLim m:val="subSup"/>

<m:naryLim m:val="undOvr"/>

</m:mathPr>

These are the default settings for the math properties element and its children.

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_MathPr) is located in §A.6.1. *End note*]

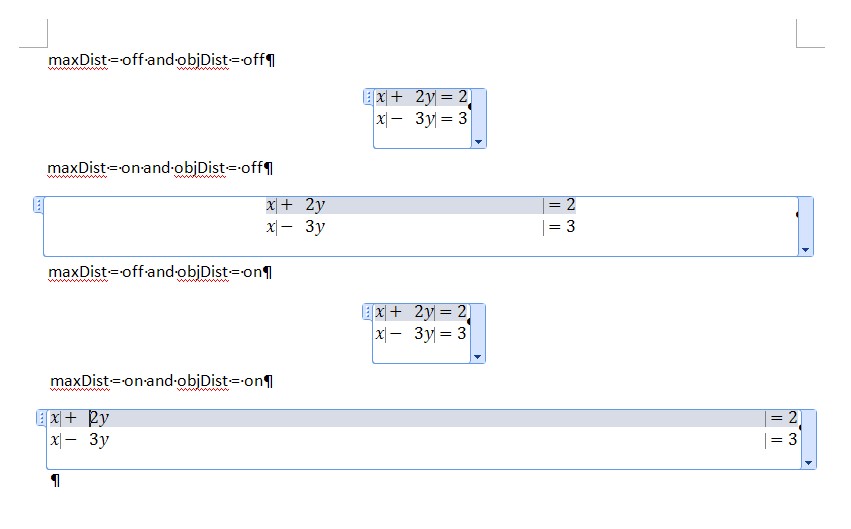
#### maxDist (Maximum Distribution)

This element specifies Array Maximum Distribution. When 1 or true, the array is spaced to the maximum width of the containing element (page, column, cell, etc.). The example image below illustrates an array expanded to fit the page, which is the containing element in this example. The maxDist option is commonly used with the objDist option. The objDist option is used to expand the distribution of mathematical text within the bounds of an array while not impacting the Array Distribution itself.

When the maxDist element is omitted, Array Maximum Distribution is off. In other words, when the element is absent, the default value of the property is 0 meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this property is applied.

[*Example*: The following diagram illustrates all four possible combinations of maxDist and objDist:



XML for the fourth example shown in the above image is:

<m:oMathPara>

<m:oMath>

<m:eqArr>

<m:eqArrPr>

<m:maxDist m:val="1" />

<m:objDist m:val="1" />

</m:eqArrPr>

<m:e>

<m:r>

<m:t>x</m:t>

</m:r>

<m:r>

<m:t>&+</m:t>

</m:r>

<m:r>

<m:t>2y&=2</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:t>x</m:t>

</m:r>

<m:r>

<m:t>&-</m:t>

</m:r>

<m:r>

<m:t>3y&=3</m:t>

</m:r>

</m:e>

</m:eqArr>

</m:oMath>

</m:oMathPara>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### mc (Matrix Column)

This element specifies a single column in a matrix m.

[*Example*: An example of this element in use is:

<m:m>

<m:mPr>

<m:mcs>

<m:mc>

<m:mcPr>

<m:count m:val="2"/>

<m:mcJc m:val="center"/>

</m:mcPr>

</m:mc>

</m:mcs>

</m:mPr>

<m:mr>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>1</m:t>

</m:r >

</m:e>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>2</m:t>

</m:r >

</m:e>

</m:mr>

<m:mr>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>3</m:t>

</m:r >

</m:e>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>4</m:t>

</m:r >

</m:e>

</m:mr>

</m:m>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_MC) is located in §A.6.1. *End note*]

#### mcJc (Matrix Column Justification)

This element specifies the justification of a matrix column (or group of matrix columns)mc. When this element is omitted, the column is centered. Whether the element is absent or present without the val attribute, the default of the val attribute is center. The matrix below has three columns. The leftmost column is left-justified, the rightmost column is right-justified, and the center column is centered:

[*Example*: A simple example of this property in use is a 2x2 matrix with both columns centered:

<m:m>

<m:mPr>

<m:mcs>

<m:mc>

<m:mcPr>

<m:count m:val="2"/>

<m:mcJc m:val="center"/>

</m:mcPr>

</m:mc>

</m:mcs>

</m:mPr>

<m:mr>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>1</m:t>

</m:r >

</m:e>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>2</m:t>

</m:r >

</m:e>

</m:mr>

<m:mr>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>3</m:t>

</m:r >

</m:e>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>4</m:t>

</m:r >

</m:e>

</m:mr>

</m:m>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the horizontal alignment of the parent element. Possible values are left, right and center. [*Example*:  <m:mcPr>  <m:mcJc m:val="center"/>  <m:count m:val="2"/>  </m:mcPr>  The possible values for this attribute are defined by the ST\_XAlign simple type (§22.9.2.18). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_XAlign) is located in §A.6.1. *End note*]

#### mcPr (Matrix Column Properties)

This element specifies the properties of the matrix column mn, including the number of columns and the type of justification. [*Example*: As an extreme example, the following matrix has two columns that are left justified (count is 2) and three columns that are right justified (count is 3).

*End example*]

[*Example*:

is represented by the following XML:

<m:m>

<m:mPr>

<m:mcs>

<m:mc>

<m:mcPr>

<m:count m:val="3"/>

<m:mcJc m:val="right"/>

</m:mcPr>

</m:mc>

</m:mcs>

</m:mPr>

<m:mr>

<m:e>

<m:r>

<m:t>1</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:t>2222</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:t>3</m:t>

</m:r>

</m:e>

</m:mr>

<m:mr>

<m:e>

<m:r>

<m:t>4444</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:t>5</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:t>6666</m:t>

</m:r>

</m:e>

</m:mr>

</m:m>

All three columns are right aligned.

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_MCPr) is located in §A.6.1. *End note*]

#### mcs (Matrix Columns)

This element specifies the collection of columns of the matrix m.[*Example*: An example of this element in use is:

<m:m>

<m:mPr>

<m:mcs>

<m:mc>

<m:mcPr>

<m:count m:val="2"/>

<m:mcJc m:val="center"/>

</m:mcPr>

</m:mc>

</m:mcs>

</m:mPr>

<m:mr>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>1</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>2</m:t>

</m:r>

</m:e>

</m:mr>

<m:mr>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>3</m:t>

</m:r >

</m:e>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>4</m:t>

</m:r >

</m:e>

</m:mr>

</m:m>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_MCS) is located in §A.6.1. *End note*]

#### mPr (Matrix Properties)

This element specifies properties of the matrix m, including the justification of the matrix and the layout of elements within the matrix.

[*Example*: is represented by the following XML:

<m:oMath>

<m:r>

<m:t>a+</m:t>

</m:r>

<m:m>

<m:mPr>

<m:baseJc m:val="top"/>

</m:mPr>

<m:mr>

<m:e>

<m:r>

<m:t>1</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:t>2</m:t>

</m:r>

</m:e>

</m:mr>

<m:mr>

<m:e>

<m:r>

<m:t>3</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:t>4</m:t>

</m:r>

</m:e>

</m:mr>

</m:m>

<m:r>

<m:t>+b</m:t>

</m:r>

</m:oMath>

This top row of the matrix is aligned with the rest of he baseline. *End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_MPr) is located in §A.6.1. *End note*]

#### mr (Matrix Row)

This element specifies a single row of the matrix m.

[*Example*: An example of this element in use is the following example, a 2x2 matrix. There are two rows; the first contains the elements 1 and 2; the second contains 3 and 4.

<m:m>

<m:mPr>

<m:mcs>

<m:mc>

<m:mcPr>

<m:count m:val="2"/>

<m:mcJc m:val="center"/>

</m:mcPr>

</m:mc>

</m:mcs>

</m:mPr>

<m:mr>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>1</m:t>

</m:r >

</m:e>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>2</m:t>

</m:r >

</m:e>

</m:mr>

<m:mr>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>3</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>4</m:t>

</m:r >

</m:e>

</m:mr>

</m:m>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_MR) is located in §A.6.1. *End note*]

#### nary (n-ary Operator Object)

This element specifies an n-ary object, consisting of an n-ary object, a base (or operand) and optional upper and lower limits. Examples of n-ary objects are: , , , .

[*Example*: The example below demonstrates an n-ary object in its proper form and XML representation:

<m:nary>

<m:naryPr>

<m:chr m:val="&#8747;"/>

</m:naryPr>

<m:sub>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>0</m:t>

</m:r>

</m:sub>

<m:sup>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>1</m:t>

</m:r>

</m:sup>

<m:e>

<m:r>

<m:t>x</m:t>

</m:r>

<m:box>

<m:boxPr>

<m:diff m:val="1"/>

</m:boxPr>

<m:e>

<m:r>

<m:t>dx</m:t>

</m:r>

</m:e>

</m:box>

</m:e>

</m:nary>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_Nary) is located in §A.6.1. *End note*]

#### naryLim (n-ary Limit Location)

This element specifies the document setting for the default placement of n-ary limits other than integrals (since integrals are most often written as subSup and other n-ary operators are most often written as undOvr), when converted from a built down form to a two-dimensional output (professional form). Limits can be either centered above and below the n-ary operator, or positioned just to the right of the operator, as in:

When this summation object is written in built down form, as in ∑\_(𝑖 = 0)^𝑛, for example, the placement of limits is ambiguous. naryLim specifies this positioning. When this element is omitted, the default placement of n-ary limits is undOvr (that is, the location of lower and upper limits). Whether the element is absent or present without the val attribute, the default of the val attribute is undOvr.

[*Example*: An example XML of this element in use is:

<m:mathPr>

<m:mathFont m:val="Cambria Math"/>

<m:brkBin m:val="before"/>

<m:brkBinSub m:val="--"/>

<m:smallFrac m:val="0"/>

<m:dispDef/>

<m:lMargin m:val="0"/>

<m:rMargin m:val="0"/>

<m:defJc m:val="centerGroup"/>

<m:wrapIndent m:val="1440"/>

<m:intLim m:val="subSup"/>

<m:naryLim m:val="undOvr"/>

</m:mathPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the default location of limits on the parent object. Possible values are subSup and undOvr.  The possible values for this attribute are defined by the ST\_LimLoc simple type (§22.1.3.8). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_LimLoc) is located in §A.6.1. *End note*]

#### naryPr (n-ary Properties)

This element specifies the properties of the n-ary object.

These include the n-ary operator character that is used (using the chr element), the shape and height of the operator (using the grow element), the location of limits (using the limLoc element) and whether empty limits display a placeholder character or are hidden (using the subHide and supHide elements).

[*Example*: N-ary operators include:∑, ∫ , ⋃, ⋀. *End example*]

[*Example*: 𝑥is represented by the following XML:

<m:nary>

<m:naryPr>

<m:chr m:val="∑"/>

<m:limLoc m:val="subSup"/>

</m:naryPr>

<m:sub>

<m:r>

<m:t>1</m:t>

</m:r>

</m:sub>

<m:sup>

<m:r>

<m:t>20</m:t>

</m:r>

</m:sup>

<m:e>

<m:r>

<m:t>x</m:t>

</m:r>

</m:e>

</m:nary>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_NaryPr) is located in §A.6.1. *End note*]

#### noBreak (No Break)

This property specifies the "unbreakable" property on the Box object box. When 1 or true, no line breaks can occur within the box. This can be important for operator emulators that consist of more than one binary operator. When this element is not specified, breaks can occur inside box. Whether the element is absent or present without the val attribute, the default of the val attribute is1meaning that this property is applied.

[*Example*: Sample XML containing this element is below. In this box, breaks are allowed.

<m:boxPr>

<m:noBreak m:val="0"/>

</m:boxPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### nor (Normal Text)

This element specifies that the run is normal text, i.e., math italics and math spacing are not applied. In a normal text run, no characters will trigger reformatting of a linear expression into a two-dimensional expression.

When the element is absent, the default value of the property is0meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this property is applied.

[*Example*: The example below illustrates three runs of normal text, along with the XML representation of the formula:

<m:r>

<m:rPr>

<m:nor/>

</m:rPr>

<m:t>rate</m:t>

</m:r>

<m:r>

<m:t>=</m:t>

</m:r>

<m:f>

<m:num>

<m:r>

<m:rPr>

<m:nor/>

</m:rPr>

<m:t>distance</m:t>

</m:r>

</m:num>

<m:den>

<m:r>

<m:rPr>

<m:nor/>

</m:rPr>

<m:t>time</m:t>

</m:r>

</m:den>

</m:f>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### num (Numerator)

This element specifies the numerator of the Fraction object f. [*Example*: The 𝑎 in :

<m:f>

<m:fPr>

<m:type m:val="skw"/>

</m:fPr>

<m:num>

<m:r>

<m:t>a</m:t>

</m:r>

</m:num>

<m:den>

<m:r>

<m:t>b</m:t>

</m:r>

</m:den>

</m:f>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OMathArg) is located in §A.6.1. *End note*]

#### objDist (Object Distribution)

This element specifies Array Object Distribution. When 1 or true, the contents of the array are spaced to the maximum width of the array object. When this element is omitted, the array does not receive object distribution. In other words, when the element is absent, the default value of the property is0meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this property is applied.

[*Example*:

is represented by the following XML:

<m:eqArr>

<m:eqArrPr>

<m:maxDist m:val="1"/>

<m:objDist m:val="1"/>

</m:eqArrPr>

<m:e>

<m:r>

<m:t>111&amp;a&amp;&amp;&amp;&amp;&amp;&amp;=&amp;&amp;c</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:t>111&amp;a&amp;&amp;+&amp;111&amp;b&amp;&amp;=&amp;111&amp;c</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:t>&amp;&amp;&amp;&amp;&amp;b&amp;&amp;=&amp;111&amp;c</m:t>

</m:r>

</m:e>

</m:eqArr>

Because object distribution and maximum distribution are applied, the equation array is expanded to fill the width that is allowed, and the columns are distributed horizontally within the object to the bounds of the object. *End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### oMath (Office Math)

This element specifies an instance of mathematical text. When used independently (not inside an oMathPara) with non-mathematical text preceding and/or following it, an independent oMath is interpreted as an inline math zone. All such math zones, including equations, expressions, arrays of equations or expressions and formulas are represented by oMath blocks. When used in a display math zone (a math paragraph, oMathPara), oMath is a container for an instance of mathematical text that starts on its own line and is not an inline math zone. When an oMath block is part of a display math zone, it is not itself an inline math zone. When an oMath block is not part of a display math zone, it is interpreted as its own inline math zone. The contents of an oMath block do not differ between display zone containers and independent inline math zones.

[*Example*:

<m:oMath>

<m:r>

<m:t>1+1=2</m:t>

</m:r>

</m:oMath>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OMath) is located in §A.6.1. *End note*]

#### oMathPara (Office Math Paragraph)

This element specifies a math paragraph, or display math zone, that contains one or more oMath elements that are in display mode. The oMath containers of a display math zone are not themselves considered inline math zones.

[*Example*:

is represented by the following XML:

<m:oMathPara>

<m:oMath>

<m:r>

<m:t>1+1=2</m:t>

</m:r>

<m:r>

<w:br/>

</m:r>

</m:oMath>

<m:oMath>

<m:r>

<m:t>2+2=4</m:t>

</m:r>

</m:oMath>

</m:oMathPara>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OMathPara) is located in §A.6.1. *End note*]

#### oMathParaPr (Office Math Paragraph Properties)

This property specifies properties of the math paragraph oMathPara, including justification jc.

[*Example*:

is represented by the following XML:

<m:oMathPara>

<m:oMathParaPr>

<m:jc m:val="right"/>

</m:oMathParaPr>

<m:oMath>

<m:r>

<m:t>1+1=2</m:t>

</m:r>

<m:r>

<w:br/>

</m:r>

</m:oMath>

<m:oMath>

<m:r>

<m:t>2+2=4</m:t>

</m:r>

</m:oMath>

</m:oMathPara>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OMathParaPr) is located in §A.6.1. *End note*]

#### opEmu (Operator Emulator)

This element specifies the Operator Emulator property on box. When 1 or true, the box and its contents behave as a single operator and inherit the properties of an operator. This means, for example, that the character can serve as a point for a line break and can be aligned to other operators. (For more details on the properties of an operator, see Unicode Technical Report #25, §3.2.2 and §3.2.3 and Unicode Technical Note #28.) Operator Emulators are often used when one or more glyphs combine to form an operator, such as==.

When the element is absent, the default value of the property is 0 meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this property is applied.

[*Example*: The following equation uses an Operator Emulator:

Its XML representation is as follows:

<m:r>

<m:t>a</m:t>

</m:r>

<m:box>

<m:boxPr>

<m:opEmu m:val="1"/>

<m:aln/>

</m:boxPr>

<m:e>

<m:r>

<m:t>==</m:t>

</m:r>

</m:e>

</m:box>

<m:r>

<m:t>b</m:t>

</m:r>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### phant (Phantom Object)

This element specifies the phantom object. This object has two primary uses: adding the spacing of the phantom base element e without displaying that base; and suppressing part of the glyph for spacing considerations.

[*Example*: In the right column below, the descender of the 𝑦 is suppressed (set to zero) while in the left column, it is shown. The presence or absence of the descender changes the shape of the .

|  |  |
| --- | --- |
| Without <m:phant> | With <m:phant> |
| <m:rad>  <m:radPr>  <m:degHide m:val="1" />  </m:radPr>  <m:deg />  <m:e>  <m:r>  <m:t>y</m:t>  </m:r>  </m:e>  </m:rad> | <m:rad>  <m:radPr>  <m:degHide m:val="1" />  </m:radPr>  <m:deg/>  <m:e>  <m:phant>  <m:phantPr>  <m:zeroDesc m:val="1" />  </m:phantPr>  <m:e>  <m:r>  <m:t>y</m:t>  </m:r>  </m:e>  </m:phant>  </m:e>  </m:rad> |

*End example*]

[*Note*: In typography, a *glyph* is the shape given in a particular typeface to a specific symbol. Most scripts share the notion of a baseline: an imaginary horizontal line on which characters rest. In some scripts, parts of glyphs lie below the baseline. The *descent* spans the distance between the baseline and the lowest descending glyph in a typeface and the part of a glyph that descends below the baseline has the name *descender*. Conversely, the *ascent* spans the distance between the baseline and the top of the glyph that reaches farthest from the baseline. *End note*].

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_Phant) is located in §A.6.1. *End note*]

#### phantPr (Phantom Properties)

This element specifies properties of the Phantom object, including whether the phantom is hidden or visible and the amount of space that is taken into account when laying out text and objects around phantoms.

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_PhantPr) is located in §A.6.1. *End note*]

#### plcHide (Hide Placeholders (Matrix))

This element specifies the Hide Placeholders property on a matrix m. When this property is on, placeholders do not appear in the matrix. If this element is omitted, placeholders do appear such that the locations where text can be inserted are made visible. In other words, when the element is absent, the default value of the property is 0 meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this property is applied.

[*Example*: The following two examples of matrices show the hidden (right matrix) and visible (left matrix) states of placeholders. The key difference between the placeholders in these matrices is that the left matrix contains visual representations for each placeholder in the matrix while the right matrix does not. However, the contents of each matrix remain the same between these two examples:

The plcHide element does not specify how a placeholder should be visually represented. It only specifies whether placeholders should or should not be visualized. The above example (left) is only one possible visual representation.

In the matrix described by the XML below, placeholders will be hidden:

<m:mPr>

<m:plcHide m:val="1"/>

<m:mcs>

<m:mc>

<m:mcPr>

<m:mcJc m:val="center"/>

<m:count m:val="3"/>

</m:mcPr>

</m:mc>

</m:mcs> </m:mPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### pos (Position)

This element specifies the position of the bar or group character in the parent object; the default is bot.

Whether the element is absent or present without the val attribute, the default of the val attribute is bot.

[*Example*: The XML representation for the mathematical overbar is:

<m:bar>

<m:barPr>

<m:pos m:val="top"/>

</m:barPr>

<m:e>

<m:r>

<m:t>a</m:t>

</m:r>

</m:e>

</m:bar>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the position of the parent element. Possible values are top and bot.  [*Example*:  <m:barPr>  <m:pos m:val="top"/>  </m:barPr>  The possible values for this attribute are defined by the ST\_TopBot simple type (§22.1.3.13). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_TopBot) is located in §A.6.1. *End note*]

#### postSp (Post-Paragraph Spacing)

This element specifies the spacing after a math paragraph, in twips. If this element is omitted, no spacing is applied after the paragraph.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the value, in twips, of the parent element.  The possible values for this attribute are defined by the ST\_TwipsMeasure simple type (§22.9.2.14). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_TwipsMeasure) is located in §A.6.1. *End note*]

#### preSp (Pre-Paragraph Spacing)

This element specifies the spacing before a math paragraph, in twips. If this element is omitted, no spacing is applied before the paragraph.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the value, in twips, of the parent element.  The possible values for this attribute are defined by the ST\_TwipsMeasure simple type (§22.9.2.14). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_TwipsMeasure) is located in §A.6.1. *End note*]

#### r (Run)

This element specifies a run of mathematical text.

[*Example*: is represented by the following XML:

<m:oMath>

<m:e>

<m:r>

<m:t>j+k</m:t>

</m:r>

</m:e>

</m:oMath>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_R) is located in §A.6.1. *End note*]

#### rad (Radical Object)

This element specifies the radical object, consisting of a radical, a base e and an optional degree deg. [*Example*:

Examples of rad are (XML shown below) and .

<m:rad>

<m:deg>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>3</m:t>

</m:r>

</m:deg>

<m:e>

<m:r>

<m:t>x</m:t>

</m:r>

</m:e>

</m:rad>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_Rad) is located in §A.6.1. *End note*]

#### radPr (Radical Properties)

This element specifies properties of the Radical object rad, including the hidden or shown state of the degree deg.

[*Example*: is represented by the following XML:

<m:rad>

<m:radPr>

<m:degHide m:val="1"/>

</m:radPr>

<m:deg/>

<m:e>

<m:r>

<m:t>b</m:t>

</m:r>

</m:e>

</m:rad>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_RadPr) is located in §A.6.1. *End note*]

#### rMargin (Right Margin)

This element specifies the right margin for math, in twips. If this element is omitted, no right margin is used. In other words, when the element is absent, the default value of the option is 0. When the element is present and the val attribute is absent, the default of the val attribute is 1440 (or 1 inch).Math margins are added to the paragraph settings for margins. If the sum of lMargin and rMargin exceed the width available, lMargin should be ignored. If rMargin exceeds the width available, a default indent of 1440 twips should be used. [*Example*:

The following XML demonstrates a rmargin setting of 1".

<m:mathPr>

<m:mathFont m:val="Cambria Math"/>

<m:brkBin m:val="before"/>

<m:brkBinSub m:val="--"/>

<m:smallFrac m:val="0"/>

<m:dispDef/>

<m:lMargin m:val="0"/>

<m:rMargin m:val="1440"/>

<m:defJc m:val="centerGroup"/>

<m:wrapIndent m:val="1440"/>

<m:intLim m:val="subSup"/>

<m:naryLim m:val="undOvr"/>

</m:mathPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the value, in twips, of the parent element.  The possible values for this attribute are defined by the ST\_TwipsMeasure simple type (§22.9.2.14). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_TwipsMeasure) is located in §A.6.1. *End note*]

#### rPr (Run Properties)

This element specifies the properties of the math run r.

[Example: is represented by the following XML:

<m:r>

<m:rPr>

<m:scr m:val="fraktur"/>

<m:sty m:val="b"/>

</m:rPr>

<m:t>F+G</m:t>

</m:r>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_RPR) is located in §A.6.1. *End note*]

#### rSp (Row Spacing (Array))

This element specifies spacing between rows of an array eqArr; it is used only when rSpRule is set to 3 (exactly; in which case the unit of measure is points) or 4 (Multiple; in which case the unit of measure is half-lines). If this element is omitted, single line spacing is used in the array and no additional spacing is used in the layout of rows. Whether the element is absent or present without the val attribute, the default of the val attribute is 0.

[*Example*: Below are three examples of the same array, with single line spacing, 1.5 line spacing and double line spacing:

The following eqArrhas rSp of 3 half-lines:

<m:eqArr>

<m:eqArrPr>

<m:rSpRule m:val="4"/>

<m:rSp m:val="3"/>

</m:eqArrPr>

<m:e>

<m:r>

<m:t>a=b+c</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:t>d+e=f</m:t>

</m:r>

</m:e>

</m:eqArr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the amount of space between columns of the parent element (for cGp/cSp) or rows (for rSp). The manner in which this value is determined depends on the setting of the rule of the parent element.  For cGp, if the rule is set to3(or "Exactly"), then the unit is interpreted as twips. If the rule is set to4(or "Multiple"), then the unit is interpreted as number of 0.5 em increments.  For cSp, this value is measured in twips. There is no corresponding cSpRule.  Forr Sp, if the rule is set to 3 (or "Exactly"), then the unit is interpreted as points. If the rule is set to 4 (or "Multiple"), then the unit is interpreted as half-lines.  The possible values for this attribute are defined by the ST\_UnSignedInteger simple type (§22.1.3.14). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_UnSignedInteger) is located in §A.6.1. *End note*]

#### rSpRule (Row Spacing Rule)

This element specifies the type of vertical spacing between columns in a matrix. The following table demonstrates possible values of rSpRule along with their definitions and examples.

|  |  |  |
| --- | --- | --- |
| **Value** | **Line spacing between rows** | **Example (non-normative)** |
| 0 | Single line gap |  |
| 1 | 1.5-line gap |  |
| 2 | 2-line gap |  |
| 3 | Exactly (rely on value of rGp, measured in points) |  |
| 4 | Multiple (rely on value of rGp, measured in lines) |  |

Whether the element is absent or present without the val attribute, the default of the val attribute is 0 (or single line gap).

[*Example*:

is represented by the following XML:

<m:m>

<m:mPr>

<m:rSpRule m:val="2"/>

</m:mPr>

<m:mr>

<m:e>

<m:r>

<m:t>1</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:t>2</m:t>

</m:r>

</m:e>

</m:mr>

<m:mr>

<m:e>

<m:r>

<m:t>3</m:t>

</m:r>

</m:e>

<m:e>

<m:r>

<m:t>4</m:t>

</m:r>

</m:e>

</m:mr>

</m:m>

Because the row spacing rule is set to 2, there is twice as much space between the matrix rows as usual.

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the type of spacing between rows and/or columns. Possible values are 0, 1, 2, 3, or 4, whose definitions are contained in the following table:   |  |  |  | | --- | --- | --- | | Value | Column/Row Gap | Example | | 0 | Single spacing gap (1 em) |  | | 1 | 1.5 spacing gap (1.5 ems) |  | | 2 | 2 spacing gap (2 ems) |  | | 3 | Exactly (for columns, rely on value of cGp, measured in twips) (for rows, rely on value of rSp, measured in points) |  | | 4 | Multiple(for columns, rely on value of cGp, measured in 0.5 em increments) (for rows, rely on value of rSp, measured in lines) |  | |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_SpacingRule) is located in §A.6.1. *End note*]

#### scr (Script)

This element describes the script applied to the characters in the run. The XML includes the Unicode value of the character (between U+0000 and U+007F), along with the script of the character. The application maps the value and script type to the appropriate Unicode range.

Whether the element is absent or present without the val attribute, the default of the val attribute is roman.

[*Example*:

<m:r>

<m:rPr>

<m:scr m:val="fraktur"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>a</m:t>

</m:r>

<m:r>

<m:rPr>

<m:scr m:val="double-struck"/>

<m:sty m:val="p"/>

</m:rPr>

<m:t>a</m:t>

</m:r>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the script type of the parent element. Possible values are double-struck, fraktur, monospace, roman, sans-serif and script.  The possible values for this attribute are defined by the ST\_Script simple type (§22.1.3.9). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_Script) is located in §A.6.1. *End note*]

#### sepChr (Delimiter Separator Character)

This element specifies the character that separates base arguments e in the delimiter object d. If this element is omitted, the default sepChr is '|'. In other words, when the element is absent, the default is for the delimiter object separator character to be U+2502 (BOX DRAWINGS LIGHT VERTICAL).

When the element is present and the val attribute is absent, the default of the val attribute is empty which means the delimiter object separator character is absent.

[*Example*: Examples of d, each with a different sepChr, are:. The following example sets COLON (:) as the separator character:

<m:dPr>

<m:sepChr val="&#0058;"/>

</m:dPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the character used by the parent element. When it is omitted, the parent uses its assigned default.  [*Example*: Accent object ã:  <m:accPr>  <m:chr m:val="&#771;"/>  </m:accPr>  *End example*]  The possible values for this attribute are defined by the ST\_Char simple type (§22.1.3.3). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_Char) is located in §A.6.1. *End note*]

#### show (Phantom Show)

This element specifies the show property of the phantom phant. When 0 or false, the phant base e is hidden. If this element is omitted, the base e is shown. Whether the element is absent or present without the val attribute, the default of the val attribute is1meaning that this property is applied.

[*Example*: In the following example, there is a phantom of the fraction in the second radical such that only the height is preserved. The fraction does not show.

<m:phantPr>

<m:show m:val="0"/>

<m:zeroDesc m:val="1"/>

</m:phantPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### shp (Shape (Delimiters))

This element specifies the shape of delimiters in the delimiter object d. Delimiters can be centered around the math axis of the mathematical text and still be made to fit the entire height of their contents (see right-hand example below), or their height and shape can be altered to exactly match their contents (see left-hand example below). These settings significantly impact the shape of the mathematical text. When this element is omitted, delimiters are 'centered'. Whether the element is absent or present without the val attribute, the default of the val attribute is centered.[*Example*: In the examples below, delimiters will be matched to the exact shape of their contents on the left and will be centered on the right:

|  |  |
| --- | --- |
| Match | Centered |
| <m:dPr>  <m:shp m:val="match"/>  </m:dPr> | <m:dPr>  <m:shp m:val="centered"/>  </m:dPr> |

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the shape of the parent element. Possible values are match and centered.  The possible values for this attribute are defined by the ST\_Shp simple type (§22.1.3.10). |

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_Shp) is located in §A.6.1. *End note*]

#### smallFrac (Small Fraction)

This element specifies a reduced fraction size display math, such that the numerator and denominator are written in script size instead of at the size of regular text.

When the element is absent, the default value of the option is0meaning that this option is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this option is applied.

[*Example*: The XML containing this element in use is:

<m:mathPr>

<m:mathFont m:val="Cambria Math"/>

<m:brkBin m:val="before"/>

<m:brkBinSub m:val="--"/>

<m:smallFrac m:val="0"/>

<m:dispDef/>

<m:lMargin m:val="0"/>

<m:rMargin m:val="0"/>

<m:defJc m:val="centerGroup"/>

<m:wrapIndent m:val="1440"/>

<m:intLim m:val="subSup"/>

<m:naryLim m:val="undOvr"/>

<m:mathPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### sPre (Pre-Sub-Superscript Object)

This element specifies the Pre-Sub-Superscript object, which consists of a base e and a subscript and superscript placed to the left of the base, as in .

[*Example*: The XML that specifies this function is:

<m:sPre>

<m:sub>

<m:r>

<m:t>1</m:t>

</m:r>

</m:sub>

<m:sup>

<m:r>

<m:t>2</m:t>

</m:r>

</m:sup>

<m:e>

<m:r>

<m:t>A</m:t>

</m:r>

</m:e>

</m:sPre>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_SPre) is located in §A.6.1. *End note*]

#### sPrePr (Pre-Sub-Superscript Properties)

This element specifies properties such as ctrlPr that can be stored on the Pre-Sub-Superscript objects Pre.

[*Example*: is represented by the following XML:

<m:sPre>

<m:sPrePr/>

<m:sub>

<m:r>

<m:t>1</m:t>

</m:r>

</m:sub>

<m:sup>

<m:r>

<m:t>2</m:t>

</m:r>

</m:sup>

<m:e>

<m:r>

<m:t>a</m:t>

</m:r>

</m:e>

</m:sPre>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_SPrePr) is located in §A.6.1. *End note*]

#### sSub (Subscript Object)

This element specifies the subscript object sSub, which consists of a base e and a reduced-size scr placed below and to the right, as in .

[*Example*: The XML that specifies this object is:

<m:sSub>

<m:e>

<m:r>

<m:t>x</m:t>

</r>

</m:e>

<m:sub>

<m:r>

<m:t>n</m:t>

</r>

</m:sub>

</m:sSub>

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_SSub) is located in §A.6.1. *End note*]

#### sSubPr (Subscript Properties)

This element specifies properties such as ctrlPr that can be stored on the Subscript object sSub.

[*Example*: is represented by the following XML:

<m:sSub>

<m:sSubPr/>

<m:e>

<m:r>

<m:t>a</m:t>

</m:r>

</m:e>

<m:sub>

<m:r>

<m:t>2</m:t>

</m:r>

</m:sub>

</m:sSub>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_SSubPr) is located in §A.6.1. *End note*]

#### sSubSup (Sub-Superscript Object)

This element specifies the sub-superscript object, which consists of a base e, a reduced-size scr placed below and to the right and a reduced-size scr placed above and to the right, as in .

[*Example*: The XML that specifies this function is:

<m:sSubSup>

<m:e>

<m:r>

<m:t>x</m:t>

</r>

</m:e>

<m:sub>

<m:r>

<m:t>m</m:t>

</r>

</m:sub>

<m:sup>

<m:r>

<m:t>n</m:t>

</r>

</m:sup>

</m:sSubSup>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_SSubSup) is located in §A.6.1. *End note*]

#### sSubSupPr (Sub-Superscript Properties)

This element specifies properties of the Sub-Superscript object, including the alignment of scripts.

[*Example*:is represented by the following XML:

<m:sSubSup>

<m:sSubSupPr>

<m:alnScr m:val="1"/>

</m:sSubSupPr>

<m:e>

<m:r>

<m:t>f</m:t>

</m:r>

</m:e>

<m:sub>

<m:r>

<m:t>100</m:t>

</m:r>

</m:sub>

<m:sup>

<m:r>

<m:t>200</m:t>

</m:r>

</m:sup>

</m:sSubSup>

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_SSubSupPr) is located in §A.6.1. *End note*]

#### sSup (Superscript Object)

This element specifies the superscript object sSup, which consists of a base e and a reduced-size scr placed above and to the right, as in .

[*Example*: The XML that specifies this object is:

<m:sSup>

<m:e>

<m:r>

<m:t>x</m:t>

</r>

</m:e>

<m:sup>

<m:r>

<m:t>n</m:t>

</r>

</m:sup>

</m:sSup>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_SSup) is located in §A.6.1. *End note*]

#### sSupPr (Superscript Properties)

This element specifies properties such as ctrlPr that can be stored on the Superscript object sSup.

[*Example*: is represented by the following XML:

<m:sSup>

<m:sSupPr/>

<m:e>

<m:r>

<m:t>a</m:t>

</m:r>

</m:e>

<m:sup>

<m:r>

<m:t>2</m:t>

</m:r>

</m:sup>

</m:sSup>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_SSupPr) is located in §A.6.1. *End note*]

#### strikeBLTR (Border Box Strikethrough Bottom-Left to Top-Right)

This element specifies the hidden or shown state of a strikethrough diagonal line from the bottom-left corner to the top-right corner of borderBox. When this element is omitted, the strikethrough is not drawn. In other words, when the element is absent, the default value of the property is0meaning that this property is not applied. When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this property is applied. When applied, a strikethrough is drawn, as in .

[*Example*:

<m:borderBox>

<m:borderBoxPr>

<m:hideTop m:val="1"/>

<m:hideBot m:val="1"/>

<m:hideLeft m:val="1"/>

<m:hideRight m:val="1"/>

<m:strikeBLTR m:val="1"/>

</m:borderBoxPr>

<m:e>

<m:r>

<m:t>abc</m:t>

</m:r>

</m:e>

</m:borderBox>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### strikeH (Border Box Strikethrough Horizontal)

This element specifies the hidden or shown state of a strikethrough horizontal line in borderBox. When this element is omitted, the strikethrough is not drawn. In other words, when the element is absent, the default value of the property is0meaning that this property is not applied. When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this property is applied. When on, a horizontal strikethrough is drawn, as in .

[*Example*:

<m:borderBox>

<m:borderBoxPr>

<m:hideTop m:val="1"/>

<m:hideBot m:val="1"/>

<m:hideLeft m:val="1"/>

<m:hideRight m:val="1"/>

<m:strikeH m:val="1"/>

</m:borderBoxPr>

<m:e>

<m:r>

<m:t>abc</m:t>

</m:r>

</m:e>

</m:borderBox>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### strikeTLBR (Border Box Strikethrough Top-Left to Bottom-Right)

This element specifies the hidden or shown state of a strikethrough diagonal line from the top-left corner to the bottom-right corner of borderBox. When this element is omitted, the strikethrough is not drawn. In other words, when the element is absent, the default value of the property is0meaning that this property is not applied. When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this property is applied. When applied, a strikethrough is drawn, as in .

[*Example*:

<m:borderBox>

<m:borderBoxPr>

<m:hideTop m:val="1"/>

<m:hideBot m:val="1"/>

<m:hideLeft m:val="1"/>

<m:hideRight m:val="1"/>

<m:strikeTLBR m:val="1"/>

</m:borderBoxPr>

<m:e>

<m:r>

<m:t>abc</m:t>

</m:r>

</m:e>

</m:borderBox>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### strikeV (Border Box Strikethrough Vertical)

This element specifies the hidden or shown state of a strikethrough vertical line in borderBox. When this element is omitted, the strikethrough is not drawn. In other words, when the element is absent, the default value of the property is0meaning that this property is not applied. When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this property is applied. When applied, a strikethrough is drawn, as in .

[*Example*:

<m:borderBox>

<m:borderBoxPr>

<m:strikeV m:val="1"/>

</m:borderBoxPr>

<m:e>

<m:r>

<m:t>abc</m:t>

</m:r>

</m:e>

</m:borderBox>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### sty (style)

This element describes the script applied to the characters in the run. The XML includes the Unicode value of the character along with the style of the character. The application maps the value and style to the appropriate Unicode range. Whether the element is absent or present without the val attribute, the default of the val attribute is i.

[*Example*:

<m:oMath>

<m:r>

<m:rPr>

<m:scr m:val="roman"/>

<m:sty m:val="b"/>

</m:rPr>

<m:t>ab</m:t>

</m:r>

</m:oMath>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the style of the parent element. Possible values are b(bold), i(italic), bi(bolditalic) and p(plain).  The possible values for this attribute are defined by the ST\_Style simple type (§22.1.3.12). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_Style) is located in §A.6.1. *End note*]

#### sub (Subscript (Pre-Sub-Superscript))

This element specifies the subscript of the Pre-Sub-Superscript object sPre. [*Example*: For example, the sub in the object is 1. An example of this element in use is:

<m:sPre>

<m:sub>

<m:r>

<m:t>1</m:t>

</m:r>

</m:sub>

<m:sup>

<m:r>

<m:t>2</m:t>

</m:r>

</m:sup>

<m:e>

<m:r>

<m:t>A</m:t>

</m:r>

</m:e>

</m:sPre>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OMathArg) is located in §A.6.1. *End note*]

#### subHide (Hide Subscript (n-ary))

This element specifies the n-ary Hide Subscript property. When 1 or true, the lower limit does not appear, as in . If this element is omitted, the lower limit appears.

When this element is omitted, the default is false. In other words, When the element is absent, the default value of the property is 0 meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this property is applied.

[*Example*: An example of this element in use is:

<m:naryPr>

<m:chr m:val="&#8747;"/>

<m:subHide m:val="1"/> </m:naryPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### sup (Superscript (Superscript object))

This element specifies the superscript of the superscript object sSup. For example, the sup in the superscript object is 𝑛.

[*Example*: An example of this element in use is:

<m:sSup>

<m:e>

<m:r>

<m:t>x</m:t>

</m:r>

</m:e>

<m:sup>

<m:r>

<m:t>n</m:t>

</m:r>

</m:sup> </m:sSup>

*End example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OMathArg) is located in §A.6.1. *End note*]

#### supHide (Hide Superscript (n-ary))

This element specifies the n-ary Hide Superscript property. When 1 or true, the upper limit does not appear, as in . If this element is omitted, the upper limit appears.

When this element is omitted, the default is false. In other words, when the element is absent, the default value of the property is 0 meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this property is applied.

[*Example*: An example of this element in use is:

<m:naryPr>

<m:chr m:val="&#8747;"/>

<m:supHide m:val="1"/> </m:naryPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### t (Text)

This element specifies the text in a math run r.

[*Example*: is represented by the following XML:

<m:oMath>

<m:e>

<m:r>

<m:t>J+K+a+j</m:t>

</m:r> </m:e>

</m:oMath>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| xml:space(Content  Contains Significant  Whitespace)  Namespace:  http://www.w3.or g/XML/1998/namespace | Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. [*Example*: Consider the following run contained within a WordprocessingML document:  <w:r>  <w:t> significant whitespace </w:t>  </w:r>  Although there are three spaces on each side of the text content in the run, that whitespace has not been specifically marked as significant, therefore it is subject to the space preservation rules currently specified in that run's scope. *End example*]  The possible values for this attribute are defined by §2.10 of the XML 1.0 specification. |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_Text) is located in §A.6.1. *End note*]

#### transp (Transparent (Phantom))

This element specifies that the phantom is transparent for spacing. This means that if the contents of the phantom are belonging to a special spacing class (such as binary operators, relational operators, differentials, etc.), the contents of that phantom are taken into consideration when laying out text. If transparency is turned off, then the contents of the phantom are ignored during layout. When this element is omitted, transparency is 0 or false. In other words, when the element is absent, the default value of the property is 0 meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied.

In the following example, transparency is not applied on the phantom around the differential term.

. The spacing is incorrect. In the following integral, the only difference is that transparency is applied:. Now the spacing is correct.

[*Example*: An example of this element in XML is:

<m:phantPr>

<m:zeroAsc m:val="1"/>

<m:zeroDesc m:val="1"/>

<m:transp m:val="1"/> </m:phantPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### type (Fraction type)

This element specifies the type of fraction f; the default is 'bar'. Whether the element is absent or present without the val attribute, the default of the val attribute is bar. Fraction types are:

Stacked Fraction:

Skewed Fraction:

Linear Fraction:

Stack Object (No-Bar Fraction):

[Example: is represented as:

<m:f>

<m:fPr>

<m:type m:val="skw"/>

</m:fPr>

<m:num>

<m:r>

<m:t>a</m:t>

</m:r>

</m:num>

<m:den>

<m:r>

<m:t>b</m:t>

</m:r>

</m:den>

</m:f>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the type of fraction. Possible values are bar(Bar Fraction), lin(Linear Fraction), noBar(No-Bar Fraction (Stack)) and skw(Skewed).  The possible values for this attribute are defined by the ST\_FType simple type (§22.1.3.4). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_FType) is located in §A.6.1. *End note*]

#### vertJc (Vertical Justification)

This element, combined with pos of groupChrPr, specifies the vertical layout of the groupChr object. Where pos specifies the position of the grouping character, vertJc specifies the alignment of the object with respect to the baseline. For example, when the group character is above the object, vertJc of top signifies that the top of the object falls on the baseline; when vertJc is set to bot, the bottom of the object is on the baseline. The table below demonstrates the four possible combinations of groupChr layout:

|  |  |  |
| --- | --- | --- |
| Pos | vertJc | **Layout** |
| top | top |  |
| top | bot |  |
| bot | top |  |
| bot | bot |  |

When the element is absent, the default value of the property is top.

When the element is present and the val attribute is absent, the default of the val attribute is bot.

[*Example*:

is represented by the following XML:

<m:oMath>

<m:r>

<m:t>a+</m:t>

</m:r>

<m:groupChr>

<m:groupChrPr>

<m:vertJc m:val="bot"/>

</m:groupChrPr>

<m:e>

<m:r>

<m:t>b+c</m:t>

</m:r>

</m:e>

</m:groupChr>

<m:r>

<m:t>+d</m:t>

</m:r>

</m:oMath>

Since the vertical justification property is set to bot on the group-character object, the bottom of the group character object is aligned with the baseline.

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the position of the parent element. Possible values are top and bot.  [Example:  <m:barPr>  <m:pos m:val="top"/>  </m:barPr>  The possible values for this attribute are defined by the ST\_TopBot simple type (§22.1.3.13). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_TopBot) is located in §A.6.1. *End note*]

#### wrapIndent (Wrap Indent)

This element specifies the indent of the wrapped line of an instance of mathematical text. The line or lines of a wrapped instance of mathematical text after the line break can either be indented by a specified amount from the left margin, or right aligned. The default indent is 1". In other words, whether the element is absent or present without the val attribute, the default of the val attribute is 1440 twips (or 1 inch).

[*Example*: The XML below demonstrates wrapIndent in use:

<m:mathPr>

<m:mathFont m:val="Cambria Math"/>

<m:brkBin m:val="before"/>

<m:brkBinSub m:val="--"/>

<m:smallFrac m:val="0"/>

<m:dispDef/>

<m:lMargin m:val="0"/>

<m:rMargin m:val="0"/>

<m:defJc m:val="centerGroup"/>

<m:wrapIndent m:val="1440"/>

<m:intLim m:val="subSup"/>

<m:naryLim m:val="undOvr"/> </m:mathPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies the value, in twips, of the parent element.  The possible values for this attribute are defined by the ST\_TwipsMeasure simple type (§22.9.2.14). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_TwipsMeasure) is located in §A.6.1. *End note*]

#### wrapRight (Wrap Right)

This element specifies the right justification of the wrapped line of an instance of mathematical text. The line or lines of a wrapped instance of mathematical text after the line break can either be indented by a specified amount from the left margin, or right aligned. If this element is present, the continuation is right aligned.

When the element is absent, the default value of the option is0meaning that this option is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this option is applied.

[*Example*: An example of this element in use is:

<m:mathPr>

<m:mathFont m:val="Cambria Math"/>

<m:brkBin m:val="before"/>

<m:brkBinSub m:val="--"/>

<m:smallFrac m:val="0"/>

<m:dispDef/>

<m:lMargin m:val="0"/>

<m:rMargin m:val="0"/>

<m:defJc m:val="centerGroup"/>

<m:wrapRight/>

<m:intLim m:val="subSup"/>

<m:naryLim m:val="undOvr"/>

</m:mathPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### zeroAsc (Phantom Zero Ascent)

This element specifies that the phantom has zero ascent. The ascent of the contents of the phantom is not taken into account during layout. When this property is omitted, the phantom does have ascent (zero ascent is not applied).

When the element is absent, the default value of the property is0meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this property is applied.

In the following example, the differential term is contained in a phantom that zero ascent. As a result, spacing is reduced between the tip of the "d" and the radical bar:. [*Example*:

<m:phantPr>

<m:zeroAsc m:val="1"/>

</m:phantPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### zeroDesc (Phantom Zero Descent)

This element specifies that the phantom has zero descent. The descent of the contents of the phantom is not taken into account during layout. When this property is omitted, the phantom does have descent (zero descent is not applied).

When the element is absent, the default value of the property is 0 meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this property is applied.

[*Example*: In the following two examples, only the second has zero descent around the "y". Note that the radical is smaller than in the first case. 

<m:phantPr>

<m:zeroDesc m:val="1"/>

</m:phantPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

#### zeroWid (Phantom Zero Width)

This element specifies that the phantom has zero width. The width of the contents of the phantom is not taken into account during layout. When this property is omitted, the phantom does have width (zero width is not applied).

When the element is absent, the default value of the property is0meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is1meaning that this property is applied.

[*Example*: In the following example, the second radical contains a phantom of the fraction a/b. The phantom has zero width, such that only the height grows to accommodate the hidden fraction: Obraz zawierający zegar, zegarek

Opis wygenerowany automatycznie

<m:phantPr>

<m:show m:val="0"/>

<m:zeroDesc m:val="1"/>

</m:phantPr>

*End example*]

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| val (Value) | Specifies a binary value for the property defined by the parent XML element.  A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute and is implied when the parent element is present.  A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present.  The possible values for this attribute are defined by the ST\_OnOff simple type (§22.9.2.7). |

[*Note*: The W3C XML Schema definition of this element’s content model (CT\_OnOff) is located in §A.6.1. *End note*]

### Simple Types

This is the complete list of simple types dedicated to Math.

#### ST\_BreakBin (Break Binary Operators)

This defines how to represent binary operators with respect to a line-wrapping break. The line can wrap before the operator or after the operator; alternately, the operator can appear both at the end of the first line and the beginning of the second.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| after(After) | When line-wrapping breaks occur on binary operators, the binary operator appears after the break (at the start of the next line). |
| before(Before) | When line-wrapping breaks occur on binary operators, the binary operator appears before the break (at the end of the first line). |
| repeat(Repeat) | When line-wrapping breaks occur on binary operators, the binary operator appears on both sides of the break (at the end of the first line and the start of the next line). |

[*Note*: The W3C XML Schema definition of this simple type’s content model (ST\_BreakBin) is located in §A.6.1. *End note*]

#### ST\_BreakBinSub (Break on Binary Subtraction)

This simple type specifies how to represent subtraction on both sides of a line-wrapping break, when the Break Binary Operators option is set to repeat. The first character represents the sign at the end of the line with the break; the second represents the sign at the start of the wrapped line. Options are --, -+ and +-.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| +-(Plus Minus) | Repetition of subtraction sign after a line-wrapping break is plus on the first line and minus on the second line. |
| -+(Minus Plus) | Repetition of subtraction sign after a line-wrapping break is minus on the first line and plus on the second line. |
| --(Minus Minus) | Repetition of subtraction sign after a line-wrapping break is minus on the first and second lines. |

[*Note*: The W3C XML Schema definition of this simple type’s content model (ST\_BreakBinSub) is located in §A.6.1. *End note*]

#### ST\_Char (Character)

This Simple Type specifies the single character used by the parent element.

[*Example*: In the following example,{𝑎} uses { and } as its enclosing characters, instead of the default ( and ).

<m:dPr>

<m:begChr m:val="{"/>

<m:endChr m:val="}"/>

</m:dPr>

*End example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type also specifies the following restrictions:

* This simple type's contents have a maximum length of1characters.

[*Note*: The W3C XML Schema definition of this simple type’s content model (ST\_Char) is located in §A.6.1. *End note*]

#### ST\_FType (Fraction Type)

Fractions can be of type bar (horizontal fraction bar), skewed ("skw" - diagonal fraction bar with kerned and vertically adjusted numerator and denominator), linear ("lin" - diagonal fraction bar, takes up exactly one line of space) and the "stack" object ("noBar").

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| bar(Bar Fraction) | Fraction with a horizontal fraction bar. |
| lin(Linear Fraction) | Fraction with slanted fraction bar, that takes up no additional vertical space. |
| noBar(No-Bar Fraction (Stack)) | Stack object, which looks like a fraction with no fraction bar. |
| skw(Skewed) | Fraction with diagonal fraction bar. |

[*Note*: The W3C XML Schema definition of this simple type’s content model (ST\_FType) is located in §A.6.1. *End note*]

#### ST\_Integer2 (Integer value (-2 to 2))

This simple type contains a value from (-2,+2) which specifies the size of the argument. The effects of each value are described by the referencing element.

This simple type's contents are a restriction of the W3C XML Schema integer datatype.

This simple type also specifies the following restrictions:

* This simple type has a minimum value of greater than or equal to-2.
* This simple type has a maximum value of less than or equal to2.

[*Note*: The W3C XML Schema definition of this simple type’s content model (ST\_Integer2) is located in §A.6.1. *End note*]

#### ST\_Integer255 (Integer value (1 to 255))

This simple type specifies an integer value. The semantics of each value are discussed by the referencing element.

This simple type's contents are a restriction of the W3C XML Schema integer datatype.

This simple type also specifies the following restrictions:

* This simple type has a minimum value of greater than or equal to1.
* This simple type has a maximum value of less than or equal to255.

[*Note*: The W3C XML Schema definition of this simple type’s content model (ST\_Integer255) is located in §A.6.1. *End note*]

#### ST\_Jc (Justification)

This Simple Type specifies the justification of Math Paragraphs. Justification of the Math Paragraph can be Left, Right, Centered, or Centered as Group.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| center(Center (Text)) | Centers each instance of mathematical text individually with respect to margins. |
| centerGroup(Centered as Group (Text)) | Justifies instances of mathematical text with respect to each other and centers the group of mathematical text (the Math Paragraph) with respect to the page. |
| left(Left Justification) | Left justification of Math Paragraph |
| right(Right) | Right Justification of Math Paragraph |

[*Note*: The W3C XML Schema definition of this simple type’s content model (ST\_Jc) is located in §A.6.1. *End note*]

#### ST\_LimLoc (Limit Location)

Limits can be in one of two positions: Under-Over (undOvr- above and below the base) and SubscriptSuperscript (subSup- positioned to the side of the base, in the position of subscripts and superscripts).

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| subSup(Subscript-Superscript location) | Limits placed to the side of the base, as opposed to directly over and under. |
| undOvr(Under-Over location) | Limits placed to the directly above and below the base, as opposed to on the side. |

[*Note*: The W3C XML Schema definition of this simple type’s content model (ST\_LimLoc) is located in §A.6.1. *End note*]

#### ST\_Script (Script)

Script can be of type Roman, Script, Fraktur, Double-Struck, Sans-Serif, or Monospace.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| double-struck(double-struck) | Double-Struck Script Type |
| fraktur(Fraktur) | Fraktur Script Type |
| monospace(Monospace) | Monospace Script Type |
| **Enumeration Value** | **Description** |
| roman(Roman) | Roman Script Type |
| sans-serif(Sans-Serif) | Sans-Serif Script Type |
| script(Script) | Script Type |

[*Note*: The W3C XML Schema definition of this simple type’s content model (ST\_Script) is located in §A.6.1. *End note*]

#### ST\_Shp (Shape (Delimiters))

Delimiters shape can be centered around the argument or matched to the shape of the argument.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| centered(Centered (Delimiters)) | Delimiters are centered around their argument. |
| match(Match) | Match shape of contents of delimiters. |

[*Note*: The W3C XML Schema definition of this simple type’s content model (ST\_Shp) is located in §A.6.1. *End note*]

#### ST\_SpacingRule (Spacing Rule)

Integer value (0 to 4), representing the type of spacing between rows.

This simple type's contents are a restriction of the W3C XML Schema integer datatype.

This simple type also specifies the following restrictions:

* This simple type has a minimum value of greater than or equal to0.
* This simple type has a maximum value of less than or equal to4.

[*Note*: The W3C XML Schema definition of this simple type’s content model (ST\_SpacingRule) is located in §A.6.1. *End note*]

#### ST\_Style (Style)

Style of math can be plain, bold, italic, or bold-italic (p, bi, i, or bi).

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| b(Bold) | Bold |
| bi(Bold-Italic) | Bold-Italic |
| i(Italic) | Italic |
| p(Plain) | Plain |

[*Note*: The W3C XML Schema definition of this simple type’s content model (ST\_Style) is located in §A.6.1. *End note*]

#### ST\_TopBot (Top-Bottom)

Possible values are top and bot.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

|  |  |
| --- | --- |
| **Enumeration Value** | **Description** |
| bot(Bottom Alignment) | Aligns the bottom of the object to the baseline of the surrounding text. |
| top(Top) | Aligns the top row of the object to the baseline of the surrounding text. |

[*Note*: The W3C XML Schema definition of this simple type’s content model (ST\_TopBot) is located in §A.6.1. *End note*]

#### ST\_UnSignedInteger (Unsigned integer)

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this simple type’s content model (ST\_UnSignedInteger) is located in §A.6.1. *End note*]