XMLmind XML Editor - Support of MathML 2.0

Hussein Shafie, XMLmind Software
<xmleditor-support@xmlmind.com>

June 30, 2026

Abstract

This document describes everything you need to know to add equations to your XML documents.

Table of Contents

1. Introduction 1
2. How to add equations to your XML documents 2
3. Using the **MathML** tool 3
    3.1. Parsing simple math expressions as MathML 5
    3.2. Sample editing session 7
    3.3. Modifying existing equations 8
    3.4. Useful tips 9
4. MathML preferences 9
    4.1. Bundled math fonts 11
A. Conformance with the MathML 2 W3C standard 11
B. Contents of the "MathML support" add-on 12
C. Customizing the MathML add-on 14
    1. Customizing the contents of the **MathML** tool 14
    2. Bundling better math fonts 14
D. Integrating MathML with document types other than DocBook 5, DITA Topic and XHTML 5 15

1. Introduction

XMLmind XML Editor natively supports MathML 2.0 *presentation markup*[1] by the means of an add-on called "**MathML support**".

> If you are interested in adding equations to your XML documents, you need to make sure that the "**MathML support**" add-on has been installed. This is done using menu item "**Options** → **Install Add-ons**".

This MathML add-on may be used to:

- Create standalone MathML documents,
- Integrate MathML presentation markup with DocBook 5, DITA topics, XHTML 5.

How, in practice, to add equations to your XML documents is described in How to add equations to your XML documents [2]. The precise contents of the add-on is detailed in Appendix B, *Contents of the "MathML support" add-on* [12].

Native support means that MathML elements are treated like any other element. That is, XMLmind XML Editor does *not* embed a specialized math editor. On the contrary, all commands, all kinds of

[1]The content markup is not supported at all. Content elements are rendered as tree views.

selection, work fine with MathML elements. However, using the generic commands requires you to learn MathML. If you don't want to do that, for example, because you just want to type a couple of equations, then simply use the **MathML** tool which is installed by the add-on just below the **Edit** tool, at the top/right of the main window. Using the MathML tool is explained in Section 3, “Using the **MathML** tool” [3].

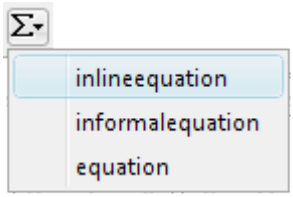
2. How to add equations to your XML documents

- If you are authoring DocBook 5 documents, insert an `equation`, `informalequation` or `inlineequation` containing a MathML `mml:math` child element into your document, then edit the contents of this `mml:math` child element normally, using the **Edit** tool and/or the **MathML** tool (see below [3]).

  In order to insert insert an `equation`, `informalequation` or `inlineequation` containing a MathML `mml:math` child element into your DocBook 5 document,

  - Use the **Edit** tool and choose one of the `equation(mathml)`, `informalequation(mathml)` or `inlineequation(mathml)` element templates.

  - OR use the "**Add MathML Equation**" found in the DocBook tool bar.

- Same approach if you are authoring a DITA topic or an XHTML 5 page.

- For all document types other than DocBook 5, DITA topics and XHTML 5 (DocBook 4, XHTML 1.*x*, etc), you must treat MathML as if it were a graphics format such as SVG or PNG.

  Unlike what happens for DocBook 5 documents, DITA topics and XHTML 5 pages, you'll not be able to directly edit the MathML elements, but if you install this MathML add-on (which depends on the add-on called "JEuclid image toolkit plug-in" — see 5 [13]), the MathML elements will be properly rendered on screen and also properly converted to formats such as HTML, PDF, RTF, etc.

  In practice, this means:

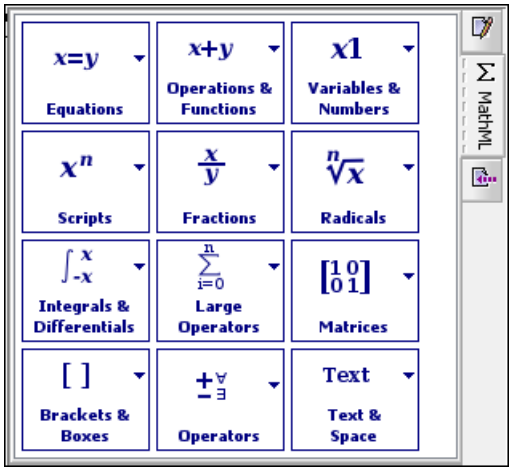
  a. Create a standalone MathML document using **File** → **New**, **MathML**, "**Inline Math**" or "**Math Block**".

  b. Reference this MathML document in the proper ``image element'':

DocBook 4 example	`<imagedata fileref="equations/equation1.mml"/>`
XHTML 1.*x* example	`<img src="equations/equation1.mml"/>`

3. Using the MathML tool

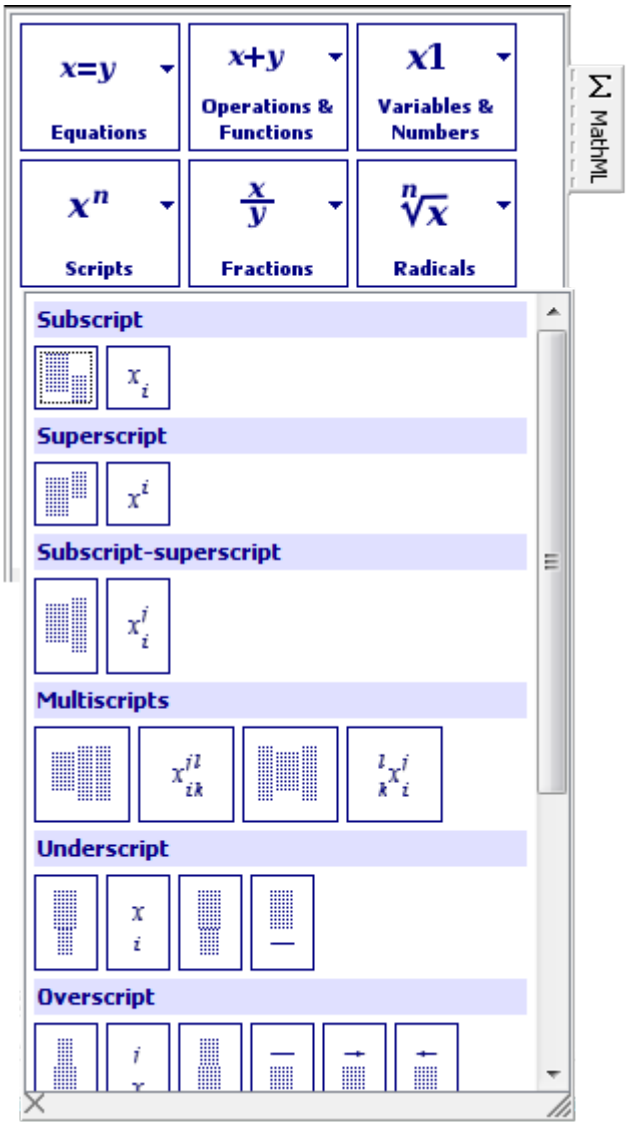
*Figure 1. The **MathML** tool*



Using the **MathML** tool does not require you to learn MathML presentation markup. However there are three points you must learn, understand and remember before using this tool:

1. Clicking on a tool in a palette replaces the explicit or implicit selection (whether it's a text or node selection) by the MathML template clicked upon. Remember that when there is no explicit text or node selection, the element containing the caret is said to be “implicitly selected”. Therefore, generally, clicking on a tool replaces the element containing the caret.

*Figure 2. Clicking on the "**x^n Scripts**" button in the **MathML** tool displays the **Scripts** palette*



2. A section in a palette[2] always starts with a completely empty template followed by one or more samples of this template.

   These samples show you how, once filled, the empty template looks like. They are also useful by themselves: you may prefer clicking on them rather than the empty template and then, use the normal text selection to replace some of the text they contain.

3. As always with XMLmind XML Editor, you need to type text in the placeholders contained in the newly inserted math template. MathML has three main text containers:

   `mi`

   represents identifiers: variable names, function names, constants, etc.

   `mn`

   represents numbers.

[2] In the above screenshot, the first section of the **Scripts** palette is **Subscript**, the second one is **Superscript**, etc

`mo`

represents operators (e.g. "+"), fences (e.g. "("), separators (e.g. ",").

Most placeholders are `mi` elements[3]. Make sure not to type a number inside an `mi` element, not to type a variable name in an `mn` element, etc. If you need to specify a number,

- click inside the `mi` placeholder and use the "**Variables & Numbers**" palette to replace it by a number,
- or, more simply, type the number then press **Ctrl**+Shift+SPACE to let XMLmind XML Editor replace the `mi` element by an `mn` element. More information about this facility in Section 3.1, “Parsing simple math expressions as MathML” [5].

Why all this trouble? After all, we are talking about presentation markup and not about content, semantic, markup. The reason is that *using the right presentation element also means using the right typesetting rules*.

Example 1: typing a single letter in an `mi` element renders this element using an italic font, while typing a longer identifier renders this `mi` element using an upright font.

Example 2: typing a curly brace "{" inside an `mo` element placed before a ``tall element'' (e.g. a fraction) will cause the curly brace to stretch vertically. Note that there is no way to force a curly brace to stretch vertically if you type this character in an `mn` or `mi` element.

Example 3: typing a "+" sign inside an `mo` element placed between two elements automatically adds the right amount of space before and after the "+" sign.

3.1. Parsing simple math expressions as MathML

Menu item **Parse Text As XML** (**MathML** → **Parse Text As MathML** for standalone MathML documents) may be used to parse as MathML simple math expressions typed inside `mi` and `mn` elements[4]. For simple equations, using this facility is much more productive than using the **MathML** tool.

For example, let's suppose you want to insert the famous $E = mc^2$ equation in your document.

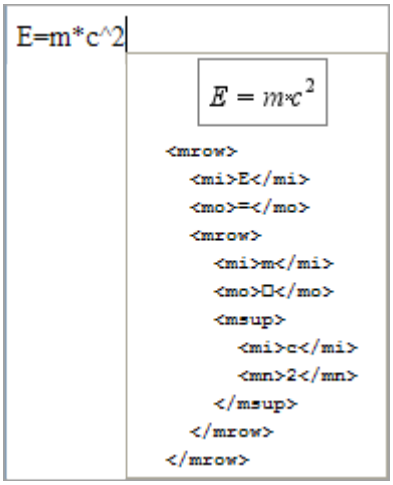
1. Use the "**Add MathML equation**" button found in the DocBook toolbar and select **inlineequation**.

2. The newly inserted element mainly contains an empty `mi` placeholder. Type "`E=m*c^2`" in this placeholder.

3. Press **Ctrl**+Shift+SPACE to invoke menu item **DocBook** → **Parse Text As XML**. Doing this displays a popup:

[3] The samples found after the empty template give you a hint on the nature of the placeholders.

[4] **Parse Text As XML** (**MathML** → **Parse Text As MathML**) is disabled (grayed) when the caret is found in the other elements which may contain text, namely: `mo`, `mtext` and `ms`.



4. Click in the popup (or press Down then Enter, like for any popup menu) in order to accept what's suggested. Et voilà, your `mi` element incorrectly containing "`E=m*c^2`" is replaced by proper MathML markup.

Of course, as we'll see it in Section 3.2, “Sample editing session” [7], this facility is also very handy when used in conjunction with **MathML** tool.

Supported expressions

The simple math expressions which can be successfully parsed by **Parse Text As XML** (**MathML** → **Parse Text As MathML**) can be quickly described as follows:

- Numbers are parsed as `mn`.
- Identifiers starting with a letter, optionally followed by letters and/or digits are parsed as `mi`. However, the following special, *case-sensitive*, identifiers are translated to a single math character:
  - Lower-case Greek letters: `alpha`, `beta`, `gamma`, `delta`, `epsilon`, `zeta`, `eta`, `theta`, `iota`, `kappa`, `lambda`, `mu`, `nu`, `xi`, `omicron`, `pi`, `rho`, `sigma`, `tau`, `upsilon`, `phi`, `chi`, `psi`, `omega`.
  - Upper-case Greek letters: `Alpha`, `Beta`, `Gamma`, `Delta`, `Epsilon`, `Zeta`, `Eta`, `Theta`, `Iota`, `Kappa`, `Lambda`, `Mu`, `Nu`, `Xi`, `Omicron`, `Pi`, `Rho`, `Sigma`, `Tau`, `Upsilon`, `Phi`, `Chi`, `Psi`, `Omega`.
  - Miscellaneous symbols: `NN` (***N***), `ZZ` (***Z***), `QQ` (***Q***), `RR` (***R***), `CC` (***C***), `emptyset` (∅), `infty` (∞).
- The only operators which are supported are: `=`, `!=` (not equal), `<`, `<=`, `>`, `>=`, `+` (unary or binary), `-` (unary or binary), `*` (translated to `⁢`), `/` (means "division", not "fraction"), `^` (exponent) and _ (here underscore means "indice"); all with their usual precedence.
- You can use parentheses for grouping. However if you do so, these parentheses will show up as `mo` elements in the MathML. If you do not want to see the parentheses in the MathML, use `{}` instead of `()`.
- Functions such as "`f(x, y, z)`" are also supported.
- Whitespace is ignored.

Examples:

Text	MathML
`2*x <= -y/0.5 <= z^i <= v_j <= w^i_j`	$2x \le -y/0.5 \le z^i \le v_j \le w^i_j$

Text	MathML
`z = x^(2*i) + y^{2*i}`	$z = x^{(2i)} + y^{2i}$
`cos(pi/2)`	$\cos(\pi/2)$
`sin(-x) = -sin(x)`	$\sin(-x) = -\sin(x)$

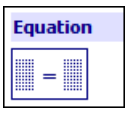
3.2. Sample editing session

First write down *on a paper* the equation you want to author using the **MathML** tool. By constantly referring to your paper while using the **MathML** tool, you'll clearly see all the top to bottom steps needed to perform your task.

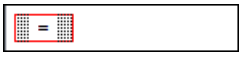
Suppose that you need to insert the following equation in a DocBook 5 document: $F = \frac{Gm_1m_2}{d^2}$.

1. Use the "**Add MathML equation**" button found in the DocBook toolbar and select **inlineequation**.

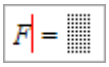
2. Click inside the newly inserted element and select the **Equations** palette in the **MathML** tool. Click on the **Equation** template:

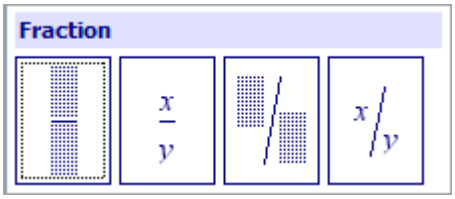
   This gives you:

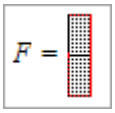
3. Type "`F`" in the left placeholder:

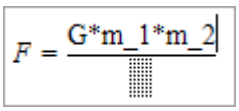
4. Click in the right placeholder then click on the first template of the "**Fractions**" palette.

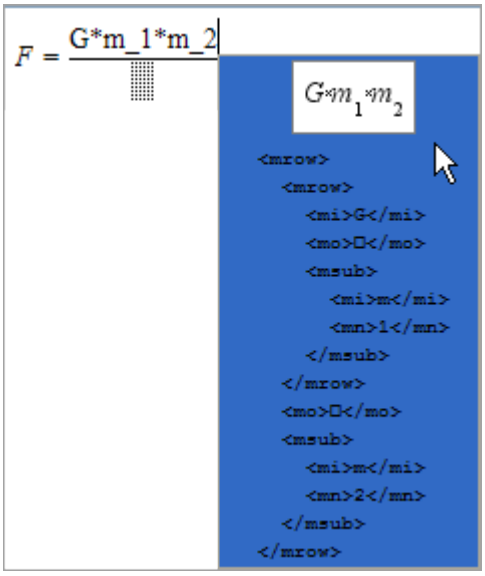
   This gives you:

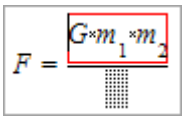
5. Type "`G*m_1*m_2`" in the top placeholder.



6. Press **Ctrl**+Shift+SPACE to invoke menu item **DocBook** → **Parse Text As XML**.



7. Press Down then Enter to accept what's suggested in the popup.



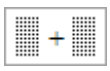
8. Type "d^2" in the bottom placeholder.

9. Press **Ctrl**+Shift+SPACE to invoke menu item **DocBook** → **Parse Text As XML**.

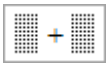
10. Press Down then Enter to accept what's suggested in the popup.

3.3. Modifying existing equations

Let's suppose you have entered the following equation: $y = \frac{2x}{3}$. After doing that, you realize that the correct equation is in fact: $y = \frac{2x}{3} + 1$. One way to correct the equation is to:

1. to explicitly select $\frac{2x}{3}$,
2. press **Ctrl**+**C** to copy the fraction to the clipboard,
3. click the  template from the "**Operations & Functions**" palette to replace the selection,
4. explicitly select the left `mi` placeholder,
5. press **Ctrl**+**V** to paste the fraction,
6. Finally, replace the right `mi` placeholder by an `mn` element containing 1.

Fortunately there is a much quicker way to do this: *wrapping* in *XMLmind XML Editor - Online Help*, rather than replacing, the explicit or implicit selection (whether it's a text or node selection). Therefore the most efficient way to correct the above equation is to proceed as follows:

1. select "`2 x 3`" normally, by dragging your mouse over this text or alternatively, use the Node Path Bar to explicitly select `m:mfrac` $\frac{2x}{3}$,
2. **Ctrl**+click (Cmd+click on the Mac) the  template in order to wrap the selected fraction,
3. Finally, replace the right `mi` placeholder by an `mn` element containing 1.

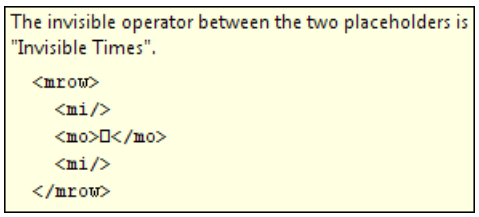
In some cases, you'll find that the text selection will not let you select the MathML element you want. For example, let's suppose you have entered $y = \sqrt{\frac{2x}{3}}$ and the correct equation is $y = \sqrt{\frac{2x}{3}} + 1$. There is no way to select the square root using the text selection.

When this is the case, you'll have to explicitly select the MathML element (for example, by clicking on `m:msqrt` in the Node Path Bar) you want to wrap and then **Ctrl**+click (Cmd +click on the Mac) on a template of a palette.

3.4. Useful tips

- When you don't understand a template, move your mouse over it and you'll see a useful “balloon help”.

  Figure 3. Tool tip for the ``Invisible Times'' binary operation

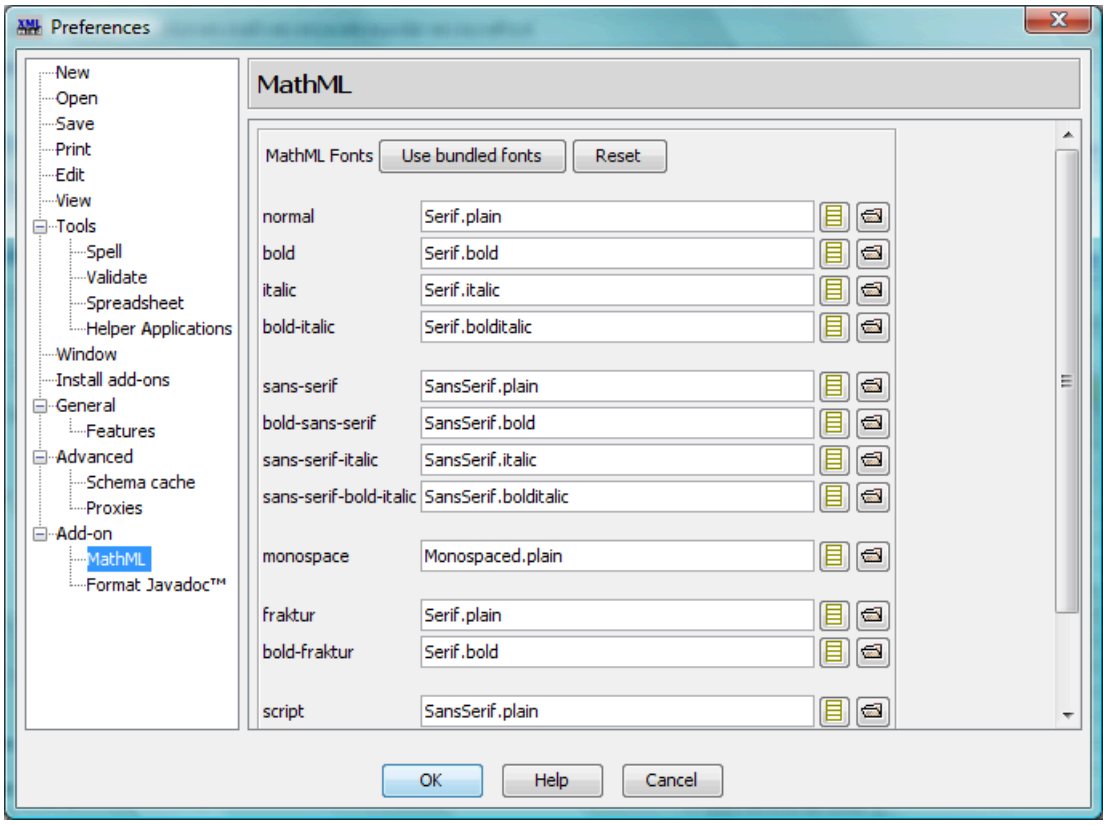
  

- The editing operations performed using a MathML palette are repeatable. That is, suffice to click elsewhere and press **Ctrl**+**A** (**Cmd**+**A** on the mac) to repeat the same replacement operation.
- The editing operations performed using a MathML palette can be recorded using the macro recorder (**Tools** → **Record Macro** → **Start**).
- The **MathML** tool has a "**Text & Space**" palette. Use it if you want to insert text which is not `mi`, `mn` or `mo`.
- If you want to edit an `mtable`, for example add a column to a matrix or add an equation to a set of equations, use the table editing commands found in the **DocBook** menu (e.g. **DocBook** → **Column** → **Insert Before**) and in the **MathML** menu (e.g. **MathML** → **Row** → **Copy**).
- If you want to insert special characters, use the **Characters** tool or use the "**Insert MathML Character by Name**" found in the **DocBook** menu and in the **MathML** menu.
- It is fairly easy to tweak the look of an equation created using the **MathML** tool. For this, you'll generally insert `mspace` elements (**Edit** → **Insert After**), wrap one or more elements into a `mphantom` or `mpadded` element (**Edit** → **Wrap**) or specify attribute values using the **Attributes** tool. Of course, doing this requires you to learn MathML presentation markup.

4. MathML preferences

Installing the MathML add-on adds a **MathML** preferences sheet to the **Preferences** dialog box (**Options** → **Preferences**). This preferences sheet allows you to specify which fonts are to be used when rendering MathML.

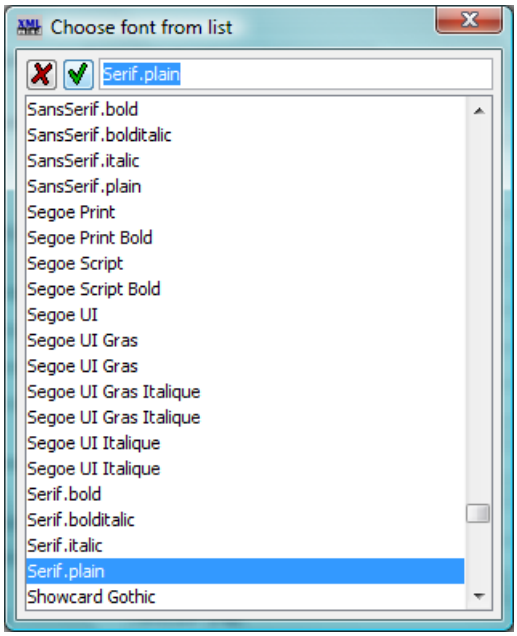
*Figure 4. The **MathML** preferences sheet*



Notice that, by default, the standard Java™ logical fonts (Serif, SansSerif, Monospaced) are also used to render the MathML elements.

There are situations where you'll want to change the above mapping. For example, on the Mac, the Serif logical font is mapped to the Times physical font and this font is intrinsically too small. In such case, you may use the "**Choose font from list**" button found at the right of each font field. This button displays a list allowing you to choose any True Type font installed on your system.

*Figure 5. Dialog box displayed by the "**Choose font from list**" button*



Alternatively, you may use the "**Choose font file**" button found at the very right of each font field. This button displays the standard file chooser which allows you to specify any `.ttf` or `.otf` font file, whether this font has been installed or not on your system.

4.1. Bundled math fonts

The MathML add-on is bundled with the following math fonts:

Logical MathML Font	Bundled Font (found in *mathml_addon_install_dir*/fonts/)	Origin
normal	STIXTwoMath-Regular.otf	STIX Fonts
bold	STIXTwoText-Bold.otf	
italic	STIXTwoText-Italic.otf	
bold-italic	STIXTwoText-BoldItalic.otf	
fraktur	typographerfraktur.medium.ttf	Free fonts from 1001 Fonts
bold-fraktur	typographerfraktur.bold.ttf	
script	tangerine.regular.ttf	
bold-script	tangerine.bold.ttf	
double-struck	sortefax.medium.ttf	

If you want to use the all above bundled fonts, simply click on the "**Use bundled fonts**" button. If you don't like the bundled fonts and want to restore the default settings, click on the **Reset** button.

If you choose to use the bundled STIX Fonts, you may have to add attribute `mathvariant="normal"` to some `mi` elements. Example, the STIX fonts cannot render the infinity symbol `U+221E` in italic (the default for all `mi` elements), therefore you'll have to change `<mi>∞</mi>` to `<mi mathvariant="normal">∞</mi>`. Note that this is consistent with MathJax which uses the same STIX fonts.

A. Conformance with the MathML 2 W3C standard

MathML support in XMLmind XML Editor passes almost all the MathML 2 tests of the W3C (Presentation section only), modulo the following limitations and specificities. Note that all advanced features (stretching of embellished operators, alignment groups and marks, etc), expect one (see below) are generally well supported.

Implementation limitations:

- The content markup is not supported at all. Content elements are rendered as tree views.
- The rendering of accents is very poor. Each MathML element, including a `mo` used as an accent, is represented on screen by a gadget[1]. This gadget has the same width, ascent and descent as the characters it contains. In XMLmind XML Editor, there is no way to make a gadget (the accent element) overlap another one (the accented element). Therefore, there is always too much space between the accent element and the accented element.

[1] A very lightweight control.

- Negative dimensions and offsets in elements such as `mpadded` are not supported.
- Limitations related to `mtable`:
  - Vertical and horizontal stretching rules are not applied to `mtd` elements containing a stretchy operator.
  - Cells acting as labels in `mlabeledtr` elements are not flushed to the right or to the left. You need to explicitly specify a `minlabelspacing` attribute for the `mtable`.
  - A percent value for the `width` attribute of an `mtable` is not supported.
  - A `fit` or percent value for the `columnwidth` attribute of an `mtable` is not supported.
  - Value `leftoverlap` of attribute `side` of an `mtable` is considered to be equivalent to value `left`. Value `rightoverlap` of attribute `side` of an `mtable` is considered to be equivalent to value `right`.
- Limitations related to `maligngroup` and `malignmark`:
  - A cell containing alignment groups is always left-aligned. That is, in such case, the `columnalign` attribute is always ignored.
  - The `decimalpoint` attribute value is honored in most cases. However, in complex cases, you may have to insert a `malignmark` element to obtain the desired alignment.
  - Empty alignment groups are ignored by the alignment algorithm.

Implementation specificities:

- Line-breaking is not implemented.
- Invisible elements such as `mphantom`, `mspace`, etc, are rendered in light gray to allow editing them. Invisible characters such as `⁢`, `⁣`, etc, are rendered using tiny vertically aligned glyphs to allow editing them.
- Comments and processing-instructions having a `math` element ancestor are not rendered at all.
- The `fontfamily` attribute of an `mglyph` element may reference the family of any font installed on the machine running XMLmind XML Editor. The `index` attribute of an `mglyph` element must specify the Unicode code point of the character represented by the `mglyph` (e.g. 97 for character 'a').

B. Contents of the "MathML support" add-on

The "MathML support" add-on contains:

1. A configuration which lets the user create standalone documents conforming to the MathML 2 *schema*[1] (presentation markup only, content markup is not supported).

   Such standalone documents are typically used the way graphics files are. XHTML example: `<img src="equations/equation1.mml"/>`.

[1] Opening a document starting with:

```
<!DOCTYPE math PUBLIC "-//W3C//DTD MathML 2.0//EN"
"http://www.w3.org/Math/DTD/mathml2/mathml2.dtd">
```

will of course cause the DTD to be used instead of the W3C XML Schema.

Note that this configuration adds not only a **MathML** menu to the menu bar, but also a **MathML** tool just below the **Edit** tool, at the top/right of the main window.

A sample MathL document is found in *`XXE_install_dir`*`/demo/mathml/sample.mml`.

2. Additional configuration elements for the DocBook 5 configuration (found in *`mathml_addon_install_dir`*`/docbook5/mathml_support.incl`). These configuration elements:

   - Let the user edit MathML elements embedded in DocBook 5 documents (by the means of elements such as `imagedata`, `equation`, `inlineequation`, etc) using the **MathML** tool.

   - Add a "**Add MathML Equation**" button to the tool bar and extends the table editing commands in order to support the `mml:mtable` element and its descendants.

   - Modify the behavior of the "**Insert special character by name**" button (found in the **Characters** tool in *XMLmind XML Editor - Online Help*; command `insertCharByName` in *XMLmind XML Editor - Commands*; keyboard shortcut **Esc** n) as follows: if the caret is inside a MathML element, the `insertCharByName` dialog box lists the names of MathML character entities (e.g. `InvisibleTimes`).

   - Replace the "**Parse Markdown as XML**" menu item (found in the configuration specific menu, e.g. the **DocBook** menu; command `unmarkdown` in *XMLmind XML Editor - Commands*; keyboard shortcut **Ctrl**+**Shift**+SPACE) by a "**Parse Text as XML**" menu item. This variant differs from `unmarkdown` in the following manner: if the caret is inside MathML elements `mml:mi` or `mml:mn`, pressing **Ctrl**+**Shift**+SPACE ignores Markdown markup and instead replaces plain text math expressions (e.g. `x+1`) by equivalent MathML elements (e.g. `<mml:mrow><mml:mi>x</mml:mi><mml:mo>+</mml:mo><mml:mn>1</mml:mn></mml:mrow>`).

   A sample DocBook 5 document containing math is found in *`XXE_install_dir`*`/demo/docbook5/sample.xml`.

3. Additional configuration elements for the DITA Topic configuration (found in *`mathml_addon_install_dir`*`/dita/mathml_support.incl`). These configuration elements add the same facilities as the above DocBook 5 customization.

   A sample DITA Topic document containing math is found in *`XXE_install_dir`*`/demo/dita/sample.dita`.

4. Additional configuration elements for the XHTML 5 configuration (found in *`mathml_addon_install_dir`*`/xhtml5/mathml_support.incl`). These configuration elements add the same facilities as the above DocBook 5 customization.

   A sample XHTML 5 page containing math is found in *`XXE_install_dir`*`/demo/xhtml5/sample.xhtml`.

5. An image toolkit similar to the one based on JEuclid.

   Note that installing this add-on also requires installing the add-on called "*JEuclid image toolkit plug-in*". That gives us *two* image toolkits which allow to convert MathML to a variety of graphics formats:

   - The “internal” image toolkit is needed to render the contents of the **MathML** tool[2].

[2] Yes, just the contents of the **MathML** tool.

- The JEuclid image toolkit is used when XML documents embedding MathML or referencing MathML files are converted to formats such as HTML, PDF, RTF, etc. The JEuclid image toolkit is especially useful when the "*Apache FOP 1.x XSL-FO processor plug-in*" add-on has been also installed because it adds “native MathML support” to the Apache FOP XSL-FO processor.

C. Customizing the MathML add-on

Normal users are not supposed to do this. The intended audience for this appendix is consultants and local gurus.

1. Customizing the contents of the MathML tool

The contents of the **MathML** tool is specified by *`mathml_addon_install_dir`*`/common/pane/mathml.pane`. Customizing the contents of the **MathML** tool consists in editing this XML file in XMLmind XML Editor, saving the changes and restarting the application.

2. Bundling better math fonts

The MathML add-on is bundled with a set of fonts [11] which are not used by default. If you want to force the use of this font set or if you want to specify an alternate font set, you'll have to specify a proper value for the "`mathml.fonts`" preference key.

The syntax of the value of this preference key is:

```
key_value --> logical_font_name ';' font_spec
              [ ';' logical_font_name ';' font_spec ]*

logical_font_name --> 'normal'
                    | 'bold'
                    | 'italic'
                    | 'bold-italic'
                    | 'sans-serif'
                    | 'bold-sans-serif'
                    | 'sans-serif-italic'
                    | 'sans-serif-bold-italic'
                    | 'monospace'
                    | 'fraktur'
                    | 'bold-fraktur'
                    | 'script'
                    | 'bold-script'
                    | 'double-struck'

font_spec --> physical_font_name
            | TTF_file_URI
```

A *`physical_font_name`* example is "`Century Schoolbook L Roman`" and not just "`Century Schoolbook L`", which is a font family.

Example C.1. Force the use of the bundled font set using the command-line

(Remove the ¬ and newline characters found after each "`;`".)

```
xxe -putpref mathml.fonts "normal;mathml-config:fonts/STIXTwoMath-Regular.otf;¬
bold;mathml-config:fonts/STIXTwoText-Bold.otf;¬
italic;mathml-config:fonts/STIXTwoText-Italic.otf;¬
bold-italic;mathml-config:fonts/STIXTwoText-BoldItalic.otf;¬
fraktur;mathml-config:fonts/typographerfraktur.medium.ttf;¬
bold-fraktur;mathml-config:fonts/typographerfraktur.bold.ttf;¬
script;mathml-config:fonts/tangerine.regular.ttf;¬
bold-script;mathml-config:fonts/tangerine.bold.ttf;¬
double-struck;mathml-config:fonts/sortefax.medium.ttf"
```

Note how the "`mathml-config:`" prefix may be used to refer to the directory where the MathML add-on has been installed. This works because the following rule has been added to *`mathml_addon_install_dir`*`/mathml_catalog.xml`:

```
<rewriteURI uriStartString="mathml-config:" rewritePrefix="." />
```

D. Integrating MathML with document types other than DocBook 5, DITA Topic and XHTML 5

Normal users are not supposed to do this. The intended audience for this appendix is consultants and local gurus.

What follows is a description of what has been done to integrate MathML with DocBook 5. Integrating MathML with other document types should be very similar.

1. File *`mathml_addon_install_dir`*`/docbook5/mathml_support.incl` contains these additional configuration elements in *XMLmind XML Editor - Configuration and Deployment*:

```
<configuration xmlns="http://www.xmlmind.com/xmleditor/schema/configuration"
  xmlns:cfg="http://www.xmlmind.com/xmleditor/schema/configuration"
  xmlns:db="http://docbook.org/ns/docbook"
  xmlns:mml="http://www.w3.org/1998/Math/MathML">

  <validate namespace="http://www.w3.org/1998/Math/MathML">❶
    <relaxng location="../common_rng/mathml2.rng" />
  </validate>

  <include location="../common/mathml.incl" />❷
  ...
  Extend the DocBook menu and toolbar here
  ...
  <elementTemplate name="mathml">❸
    <equation xmlns="http://docbook.org/ns/docbook">
      <title></title>
      <mml:math xmlns:mml="http://www.w3.org/1998/Math/MathML"
                display="block">
        <mml:mi></mml:mi>
      </mml:math>
    </equation>
  </elementTemplate>
```

```
</configuration>
```

❶ Dynamically merge the RELAX NG schema of MathML presentation markup (`mathml2.rng`) with the stock RELAX NG schema of DocBook 5 (`docbook.rng`). See below [17].

❷ Include the MathML configuration.

❸ Among other extensions, specify one or more templates for elements embedding MathML elements.

2. The above file is referenced in the stock DocBook 5 configuration (*`XXE_install_dir`*`/addon/config/docbook5/docbook5.xxe`) as follows:

```
<!-- Do not report an error if the "MathML support" add-on
     has not been installed. -->
<include location="---❶mathml-config:docbook5/mathml_support.incl❷" />
```

❶ The "`---`" prefix before an URL instructs **XXE** to silently skip the inclusion when the URL cannot be successfully resolved.

❷ URL `mathml-config:docbook5/mathml_support.incl` resolves to *`mathml_addon_install_dir`*`/docbook5/mathml_support.incl` because XML catalog *`mathml_addon_install_dir`*`/mathml_catalog.xml` contains the following rule:

```
<rewriteURI uriStartString="mathml-config:" rewritePrefix="." />
```

3. File *`mathml_addon_install_dir`*`/docbook5/mathml_support.imp` contains the following additional CSS rules:

```
@import url(../common/mathml.css);❶

@namespace db5 "http://docbook.org/ns/docbook";

db5|imagedata:contains-element(mml|math),❷
db5|inlineequation > mml|math,
db5|informalequation > mml|math,
db5|equation > mml|math {
    content: "";
}

db5|inlineequation > mml|math {
    /* Force inline-block whatever mml|math/@display. */
    display: view("com.xmlmind.xmleditext.mathml.MathElementViewFactory",
                  inline-block);
}
```

❶ Import the CSS stylesheet for MathML elements.

❷ Suppress the generated content normally used to style the `imagedata`, `inlineequation`, etc, DocBook 5 elements.

4. The above file is referenced in the stock DocBook 5 CSS stylesheet (*`XXE_install_dir`*`/addon/config/docbook5/css/docbook5.css`) as follows:

```
/* Do not report an error if the "MathML support" add-on
   has not been installed. */
@import url(---mathml-config:docbook5/mathml_support.imp);
```

How the MathML schema is dynamically merged with the DocBook 5 schema

a. The RELAX NG schema of DocBook 5 is declared in the stock DocBook 5 configuration *XXE_install_dir*/addon/config/docbook5/docbook5.xxe.

```
<relaxng location="rng/V5.0/docbook.rng" />
```

b. The `validate` configuration element allows to merge one or more auxiliary DTDs or schemas to the main DTD or schema.

```
<validate namespace="http://www.w3.org/1998/Math/MathML">
  <relaxng location="../common_rng/mathml2.rng" />
</validate>
```

You'll find:

- the W3C XML Schema for MathML (both presentation and content markup) in *mathml_addon_install_dir*/standalone/xsd/mathml2.xsd,
- the RELAX NG schema for MathML (presentation markup only) in *mathml_addon_install_dir*/common_rng/mathml2.rng,
- the RELAX NG schema Compact Syntax for MathML (presentation markup only) in *mathml_addon_install_dir*/common/pane/mathml2.rnc.

  Both `mathml2.rng` and `mathml2.rnc` have been translated from the MathML DTD using Trang[1].

However nothing forces you to merge schemas of the same kind. More information in *XMLmind XML Editor - Configuration and Deployment*.

[1] Multi-format schema converter based on RELAX NG.