FORMULATOR v3.8

MathML Suite

Mastering MathML
Presentation Markup

Published By
Hermitech, Laboratory of Mathematical and Modeling Software,
Zhytomyr, UKRAINE

E-mail: *info@mmlsoft.com*
Web: *http://www.mmlsoft.com*

Mastering MathML Presentation Markup

Mathematical templates in Formulator provide dual facilities to edit formulas. In accordance with the MathML approach to mathematics coding, there are "Presentation" and "Content" templates. The first group of templates is of special interest to users having publishing needs and has a lot of presentation abilities: fractions, radicals, sums, integrals, products, matrices, various types of brackets and braces, and many other templates. The second group is oriented on mathematical semantics and is extremely useful when the meaning of the entered formula is critical.

A group of Presentation mathematical templates comprises:

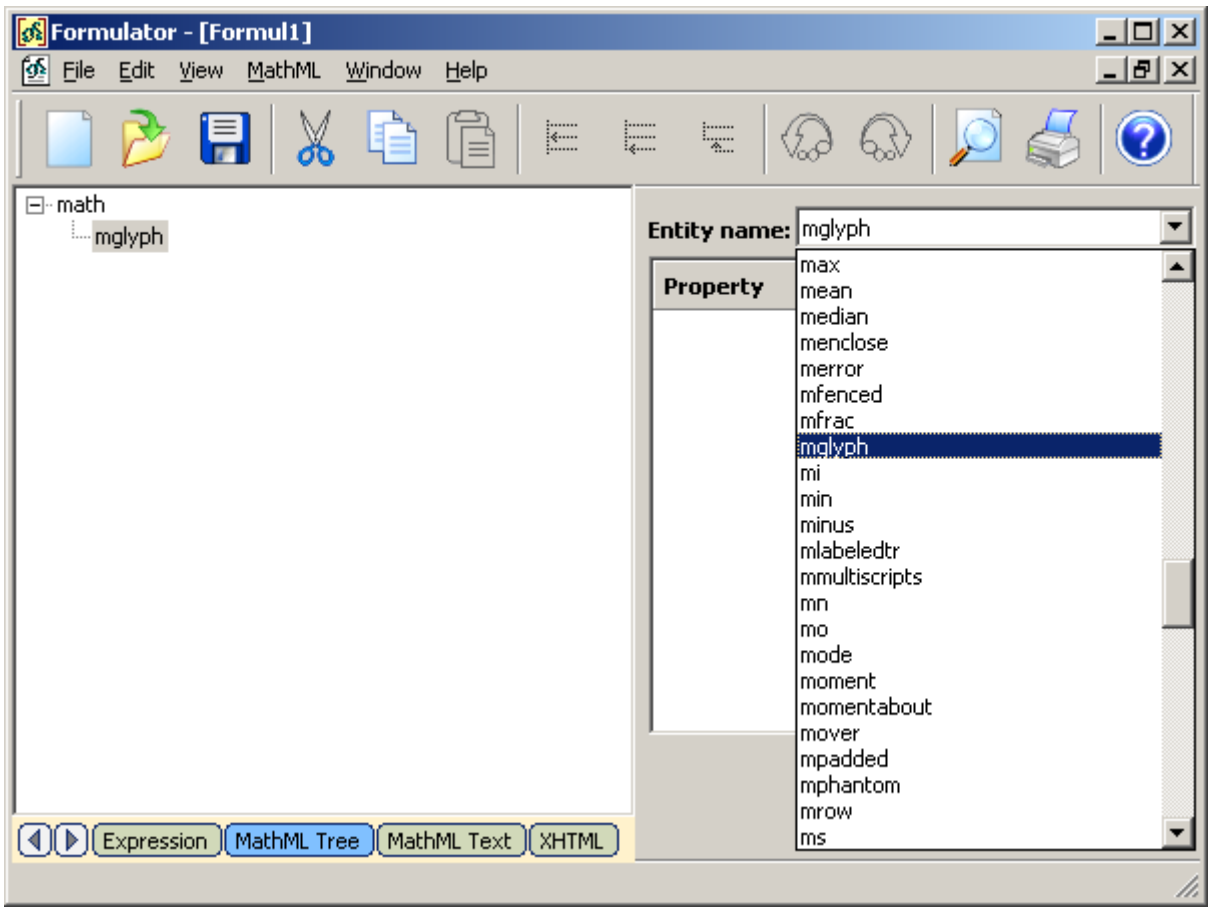
- relational and logical symbols
- spaces templates
- operator symbols
- arrow symbols
- set theory symbols
- special constants
- miscellaneous symbols
- Greek characters (lowercase)
- Greek characters (uppercase)
- differentiation templates
- fence templates
- fraction and radical templates
- subscript and superscript templates
- summation templates
- integral templates
- underbar and overbar templates
- labeled arrow templates
- products and set theory templates
- table templates

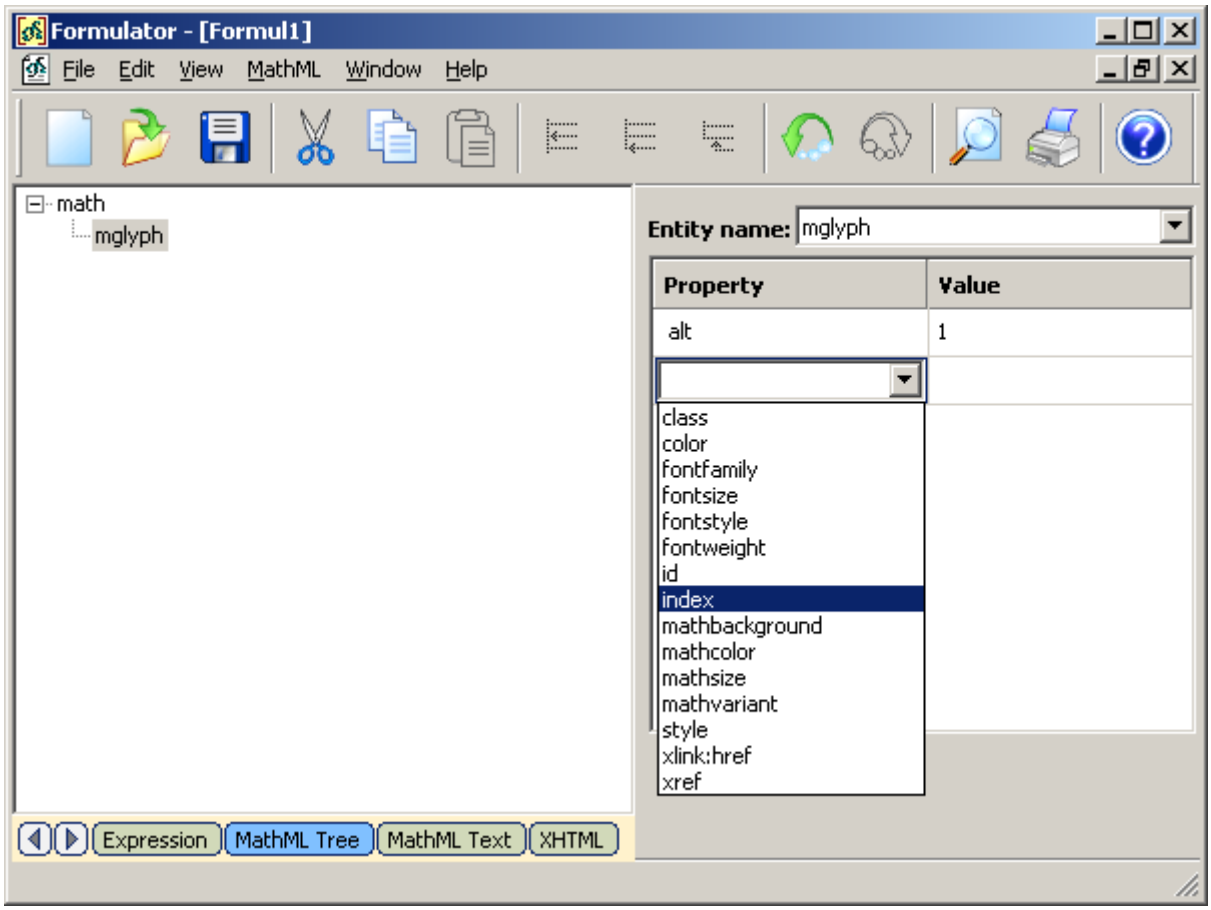
box templates

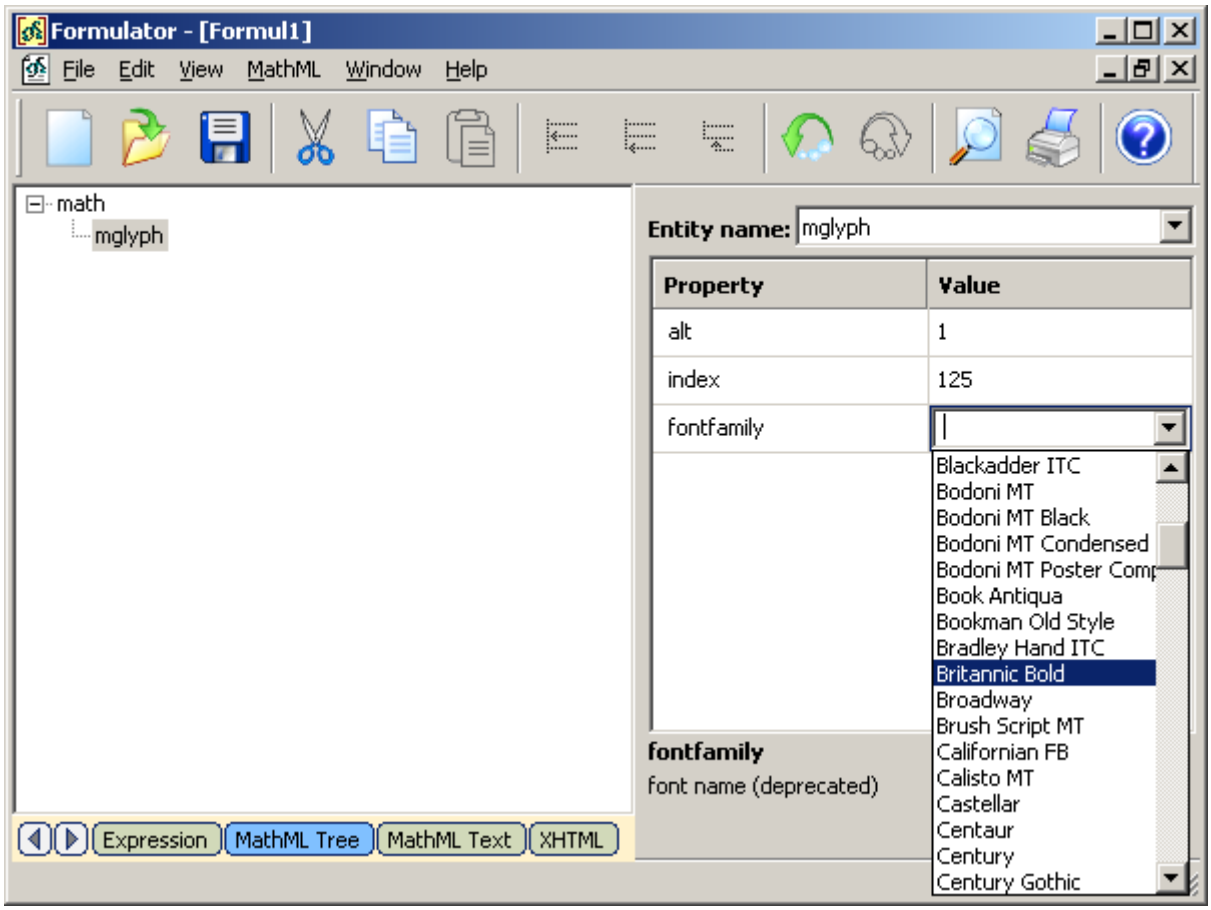
Token Elements

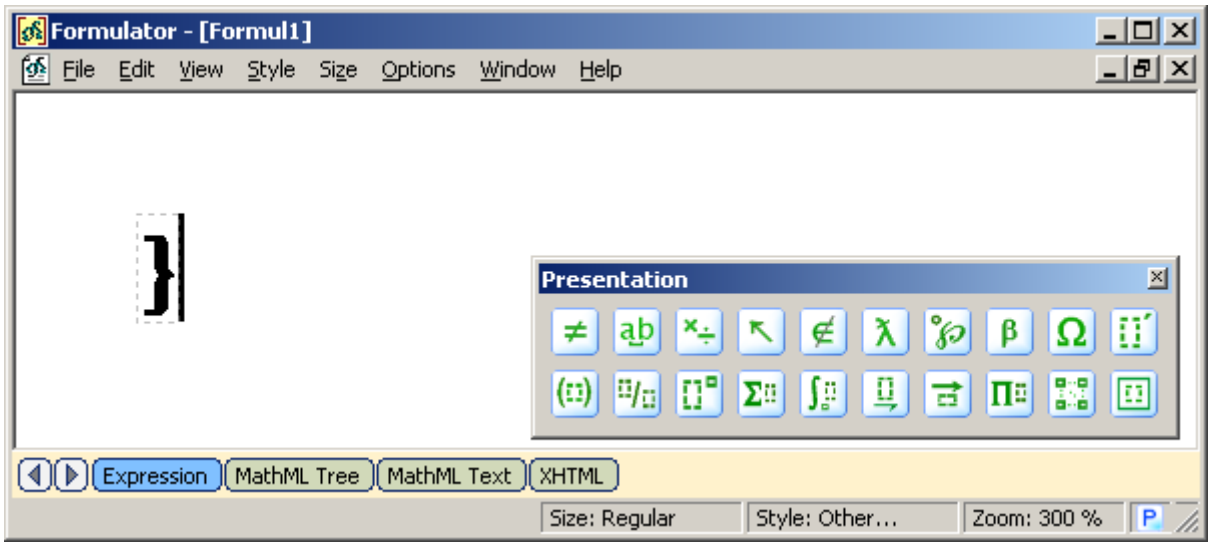
These elements can be just typed and their kind will be determined automatically if the current style is 'Math'. If a user exactly knows what kind of token elements is needed then a Style menu item can be used to specify the kind (mi, mo, mn). Besides, there is a toolbar  for "mo" equivalents of space elements.

There are token elements that neither will be detected automatically when typing, nor have a special toolbar (ms, mspace, mglyph). They can be edited manually, using "MathML Tree" or "MathML Text" options. See the next figures for the explanation.


Formulator - [Formul1]
File Edit View MathML Window Help
math
mglyph
Entity name: mglyph
Property
max
mean
median
menclose
merror
mfenced
mfrac
mglyph
mi
min
minus
mlabeledtr
mmultiscripts
mn
mo
mode
moment
momentabout
mover
mpadded
mphantom
mrow
ms
Expression
MathML Tree
MathML Text
XHTML


Formulator - [Formul1]
File Edit View MathML Window Help
math
mglyph
Entity name: mglyph
Property
Value
alt
1
class
color
fontfamily
fontsize
fontstyle
fontweight
id
index
mathbackground
mathcolor
mathsize
mathvariant
style
xlink:href
xref
Expression
MathML Tree
MathML Text
XHTML

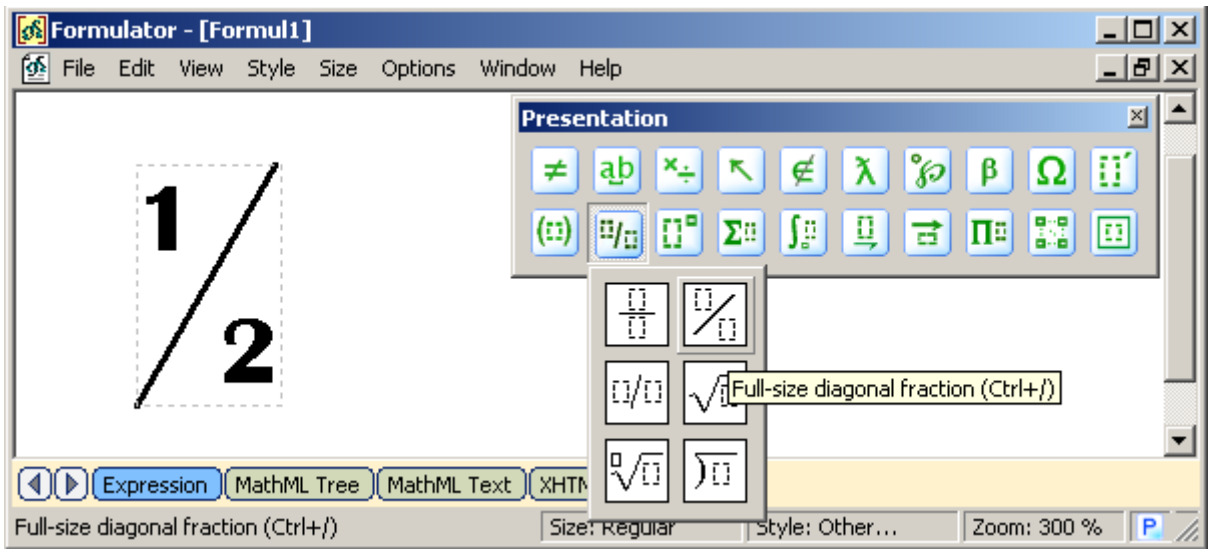


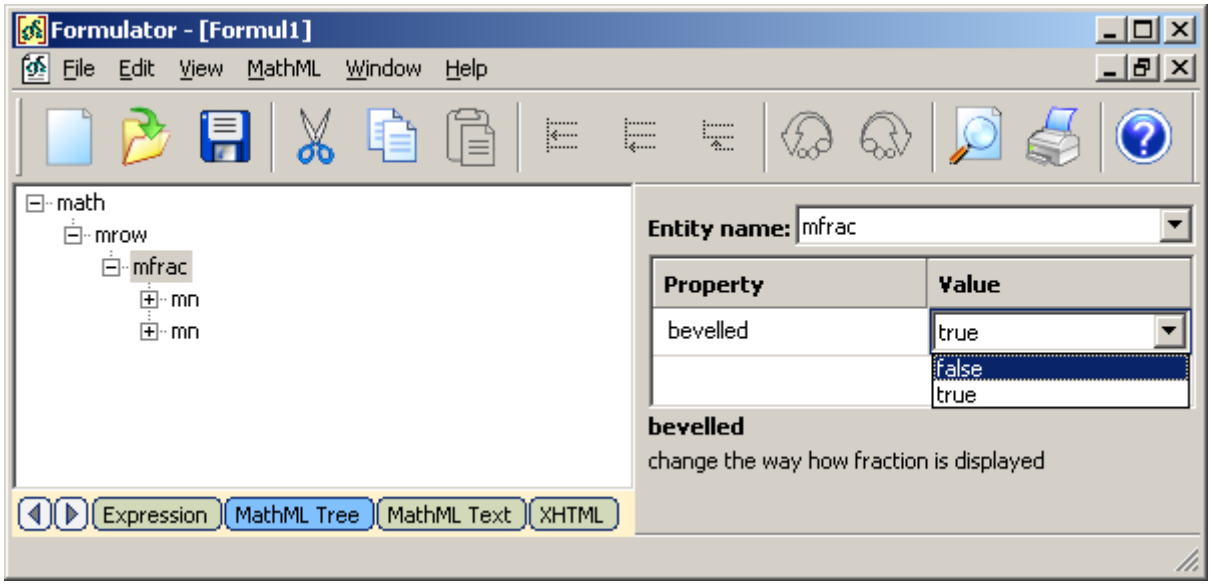


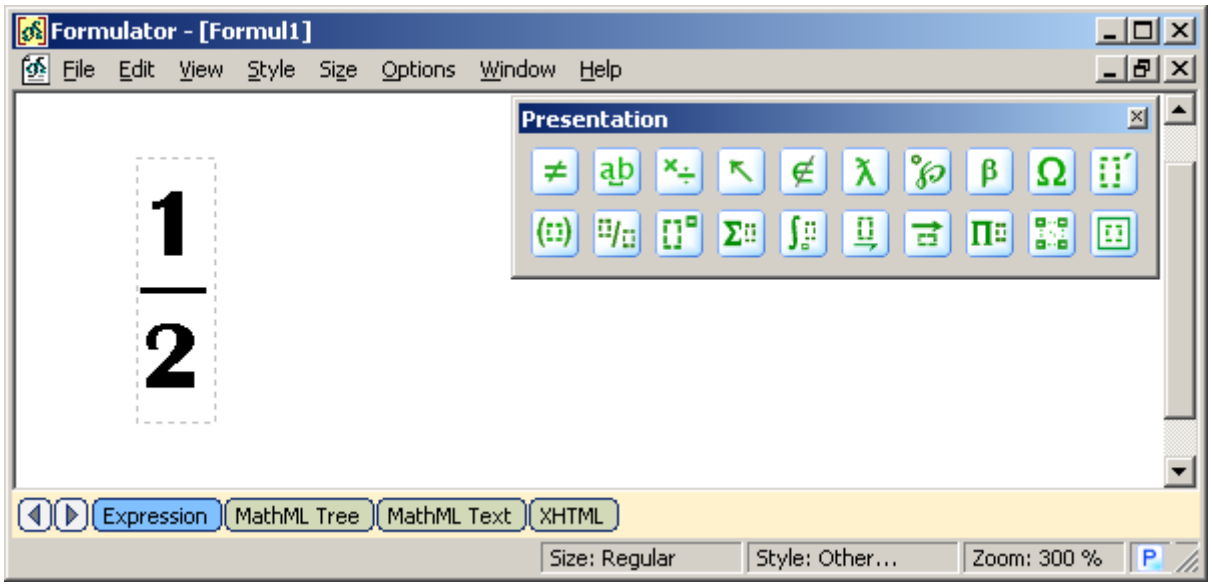
Fractions and roots

The "mfrac" element is used for fractions; the "msqrt" element is used for square roots, while the "mroot" element is used to draw radicals with indices.

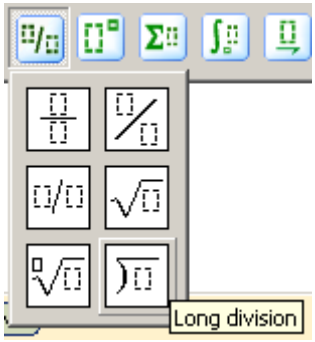
Fractions and roots can be inserted via toolbar. There are several buttons which take into account different available forms of fences and roots. Fine tuning or specific editing features can be accessed via the "MathML Tree" and "MathML Text" options. For example, the following figures show how to insert the "bevelled" version of a fraction and then convert it to a simple variant using "MathML Tree" editing mode.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Presentation
Full-size diagonal fraction (Ctrl+/)
Expression MathML Tree MathML Text XHTM
Full-size diagonal fraction (Ctrl+/)
Size: Regular
Style: Other...
Zoom: 300 %


Formulator - [Formul1]
File Edit View MathML Window Help
math
mrow
mfrac
mn
mn
Entity name: mfrac
Property
Value
bevelled
true
false
true
bevelled
change the way how fraction is displayed
Expression MathML Tree MathML Text XHTML


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Presentation
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Other...
Zoom: 300 %

Note that the button for the long division is also placed into this toolbar, but used another scheme of MathML encoding, the so-called, "enclosing notation" (menclose).

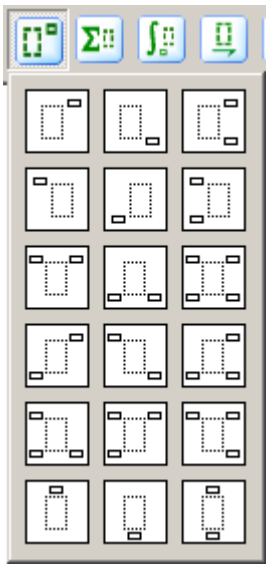


Scripts and Limits

These are the elements of the "Scripts and Limits" group:

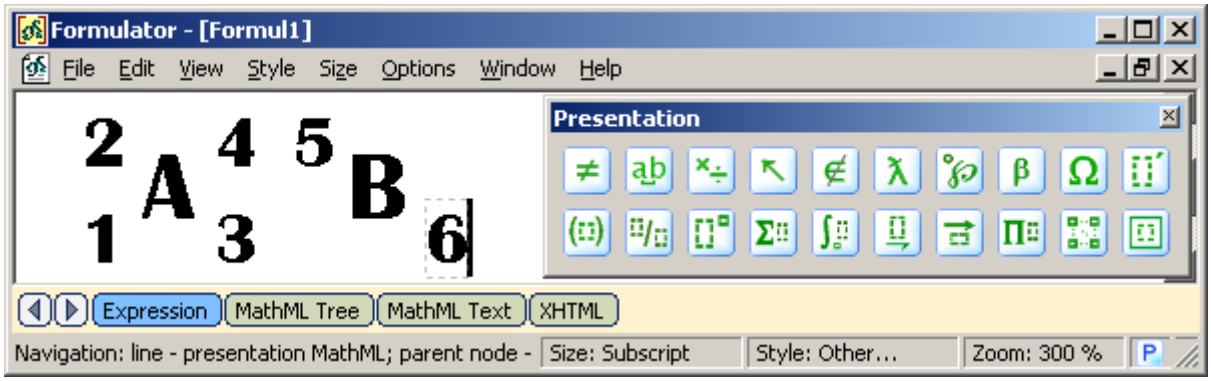
msub	attach a subscript to a base
msup	attach a superscript to a base
msubsup	attach a subscript-superscript pair to a base
munder	attach an underscript to a base
mover	attach an overscript to a base
munderover	attach an underscript-overscript pair to a base
mmultiscripts	attach prescripts and tensor indices to a base

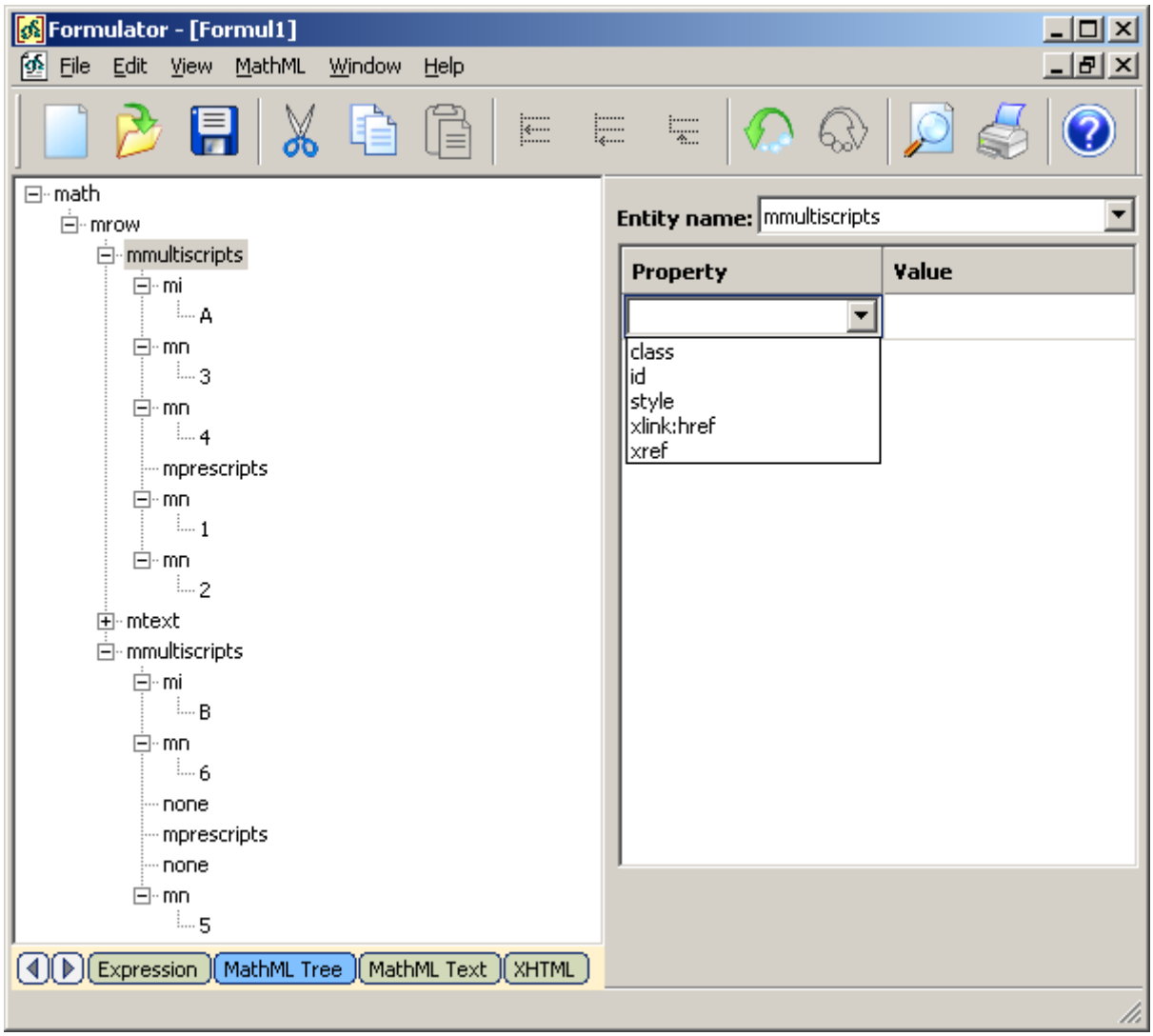
MathML Weaver works with script and limit schemata via  toolbar:



Simple elements of this schemata are implemented as partial cases (the first buttons line — msub, msup, msubsup; the last buttons line — munder, mover, munderover), but the most powerful MathML element here is "mmultiscripts", since it is able to encode all the complicated parent-child slots relations of the script and limit schemata.

The next two figures show how the "mmultiscripts" forms the equivalent MathML tree for the different cases of scripts. Note application of the <none/> and <mprescripts/> elements.





Enclose Expression Inside Notation

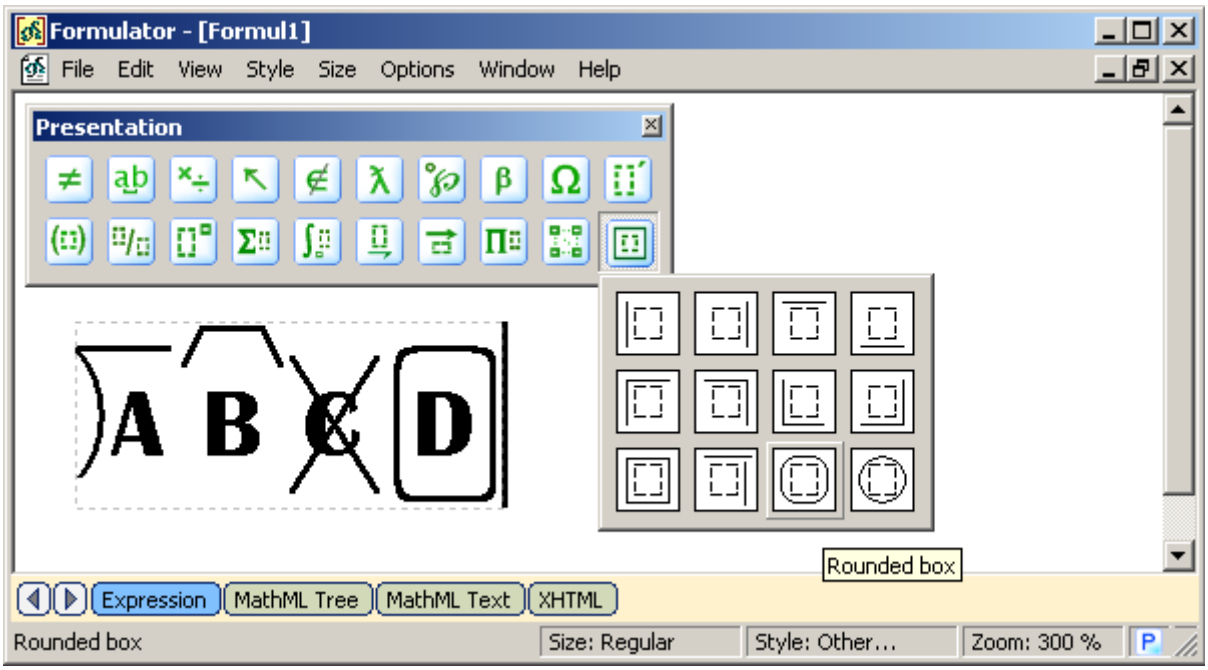
The "menclose" element renders its content inside the enclosing notation specified by its notation attribute. According to the W3C Recommendation (Mathematical Markup Language (MathML) Version 2.0), "the values allowed for notation are open-ended". So, that is a place for handling encoding variants which are not directly specified in MathML 2.0.

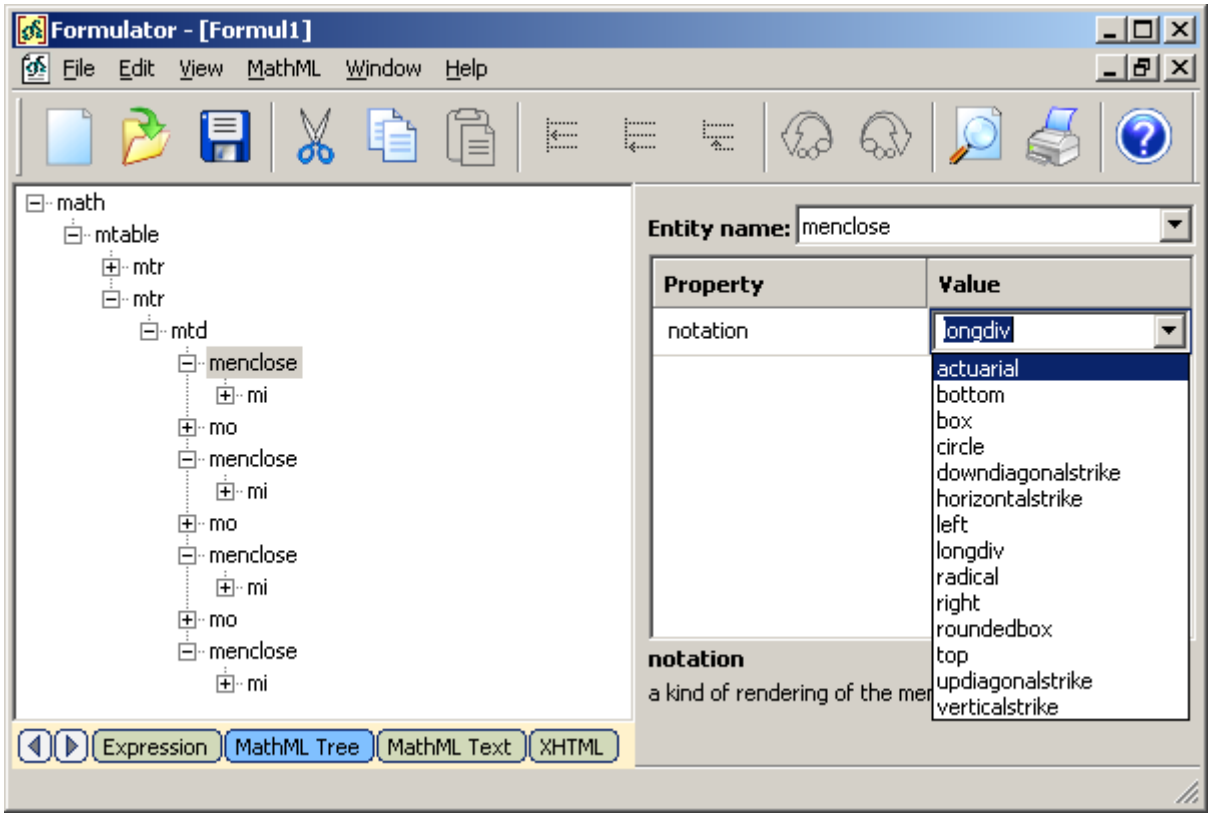
MathML Weaver benefits from the "menclose" element by implementing both the proposed by W3C values of the "notation" attribute (longdiv, actuarial, radical, box, roundedbox, circle, left, right, top, bottom, updiagonalstrike, downdiagonalstrike, verticalstrike,

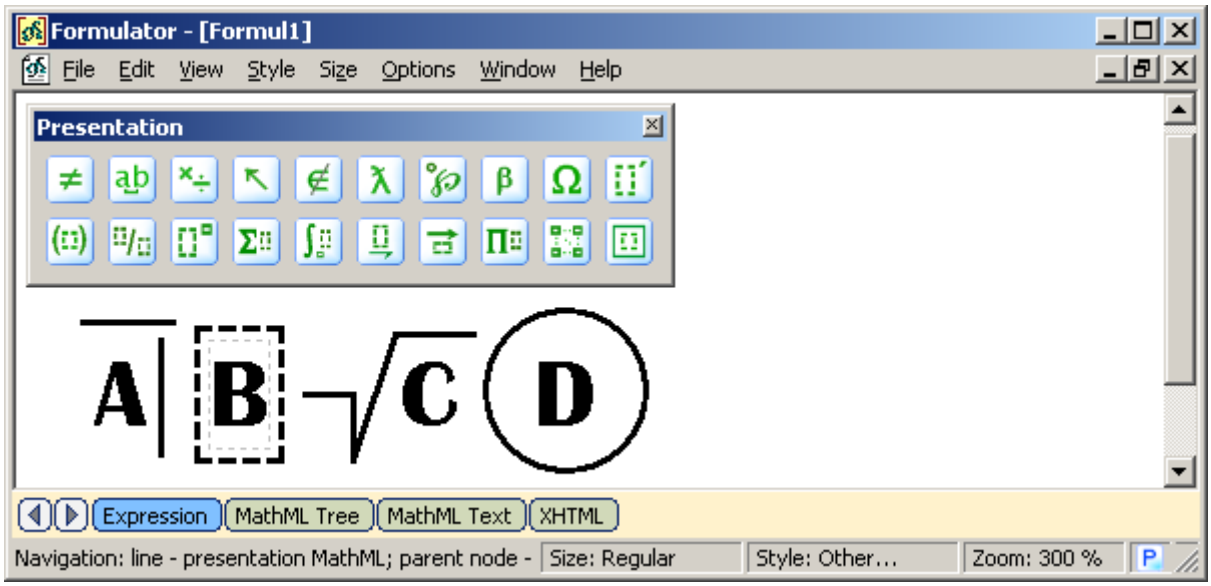
horizontalstrike) and its own values (joint-status, top-left, top-right, bottom-left, bottom-right, box-dashed). There are several toolbars for having "menclose" element, since buttons are grouped according more to their sense than to their MathML equivalent. A user can find "menclose" elements in such toolbars as:

- ✓ fractions and radicals ( );
- ✓ underbars and overbars ( );
- ✓ boxes ( ).

See the following figures for example of using "menclose" elements. We insert several formulas using "menclose" elements and then via WYSIWYG-style editing on the "MathML Tree"page we change values of the "notation" attribute. Results are shown by switching again on the "Expression" page.

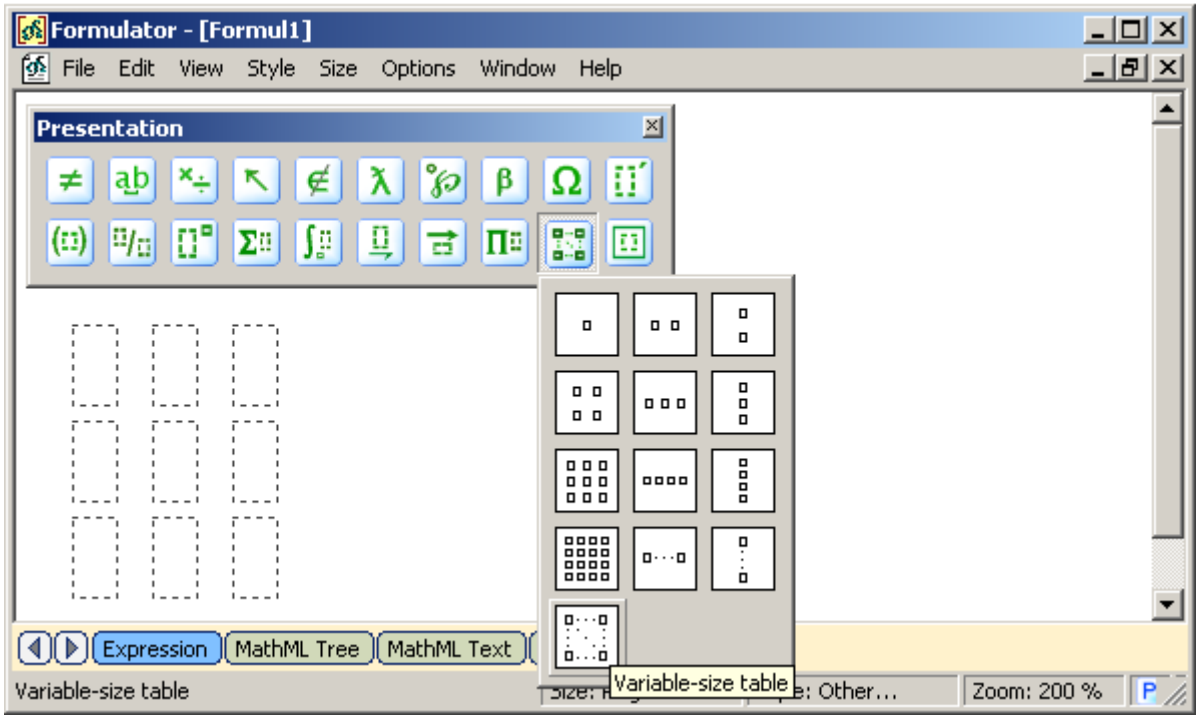




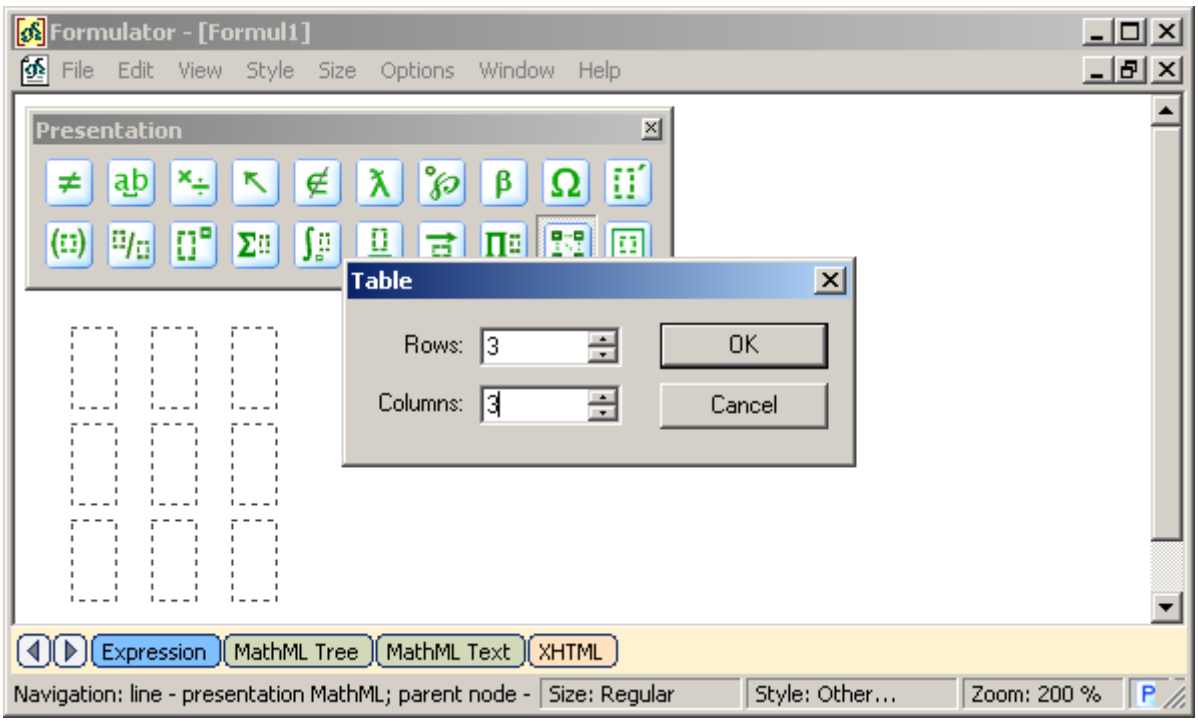


Tables and Matrices

A matrix or table is specified using the "mtable" element. It can be accessed via  toolbar:

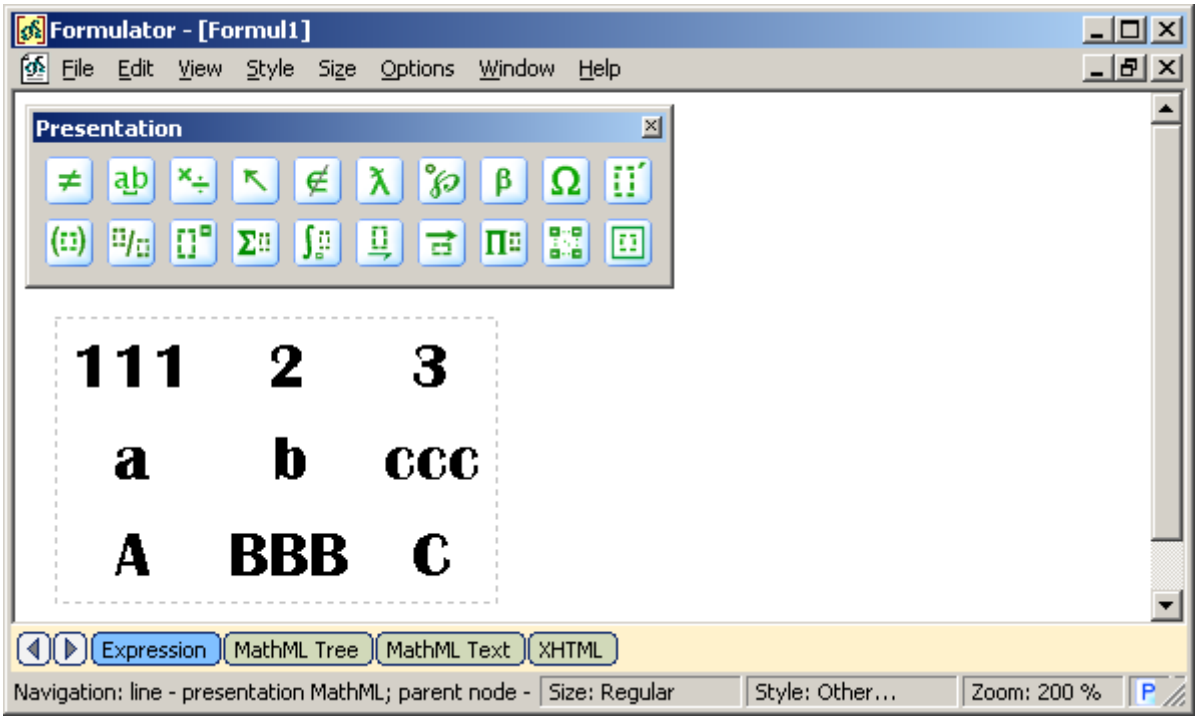


There are 12 buttons for simple cases of a matrix and one button that requests a user to enter the number of rows and columns:



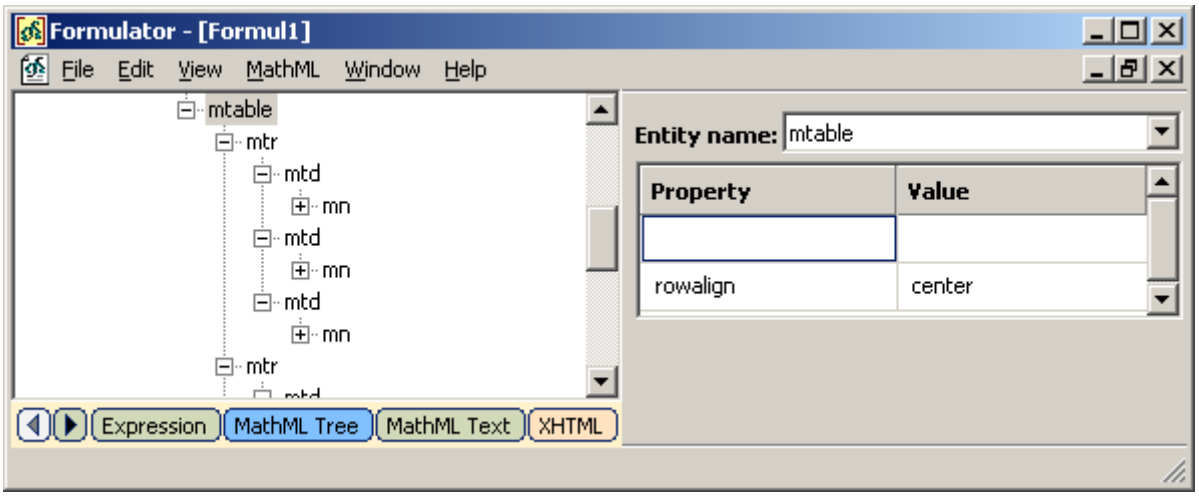
After inserting of a table or a matrix a user still has a chance to change its appearance via fine tuning of attributes for the “mtable”, “mtr” and “mtd” elements. The next example shows how to alter alignments and framing of all the table and of some of its cells.

1. Enter a table and fill it with values.

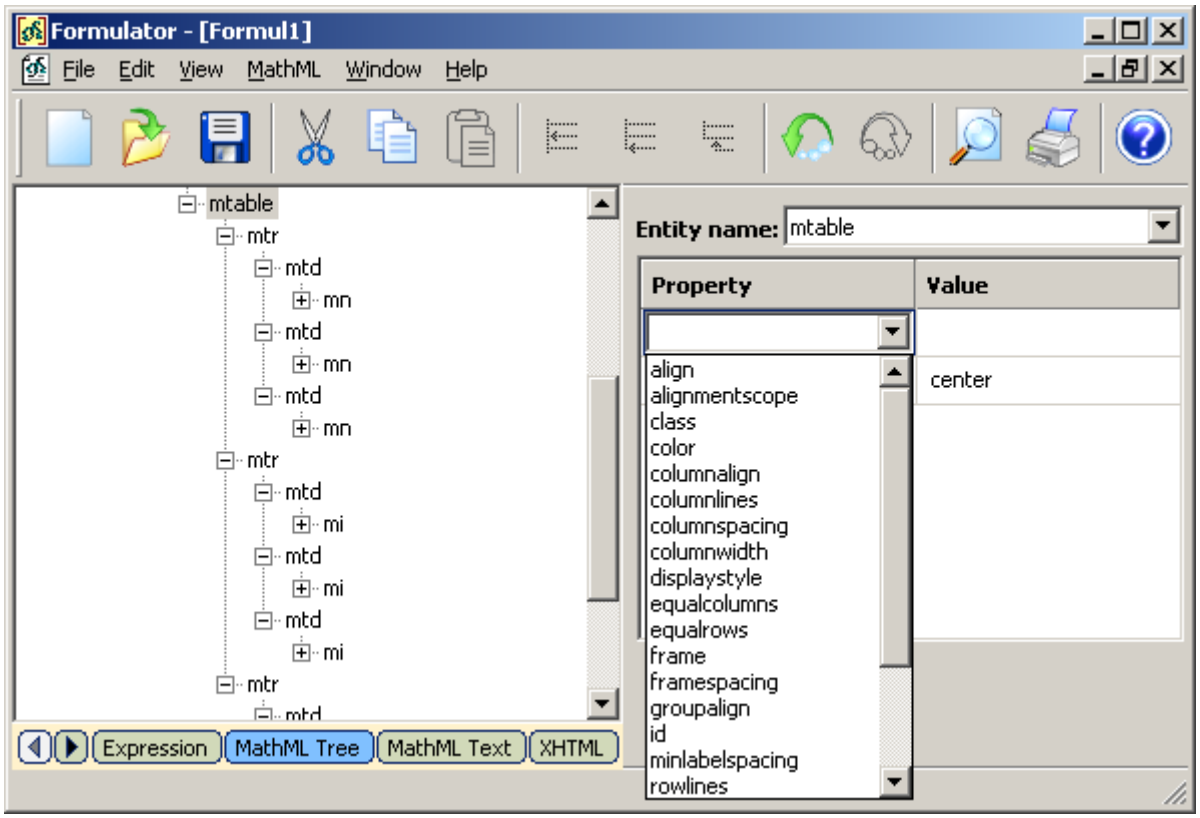


2. Switch to the "MathML Tree", select "mtable" element on the left side of the document (note that in multiline expressions we don't need the *outer* "mtable" element that is used for encoding several lines as rows of a table). Now the right side indicates current attributes of the "mtable" element.

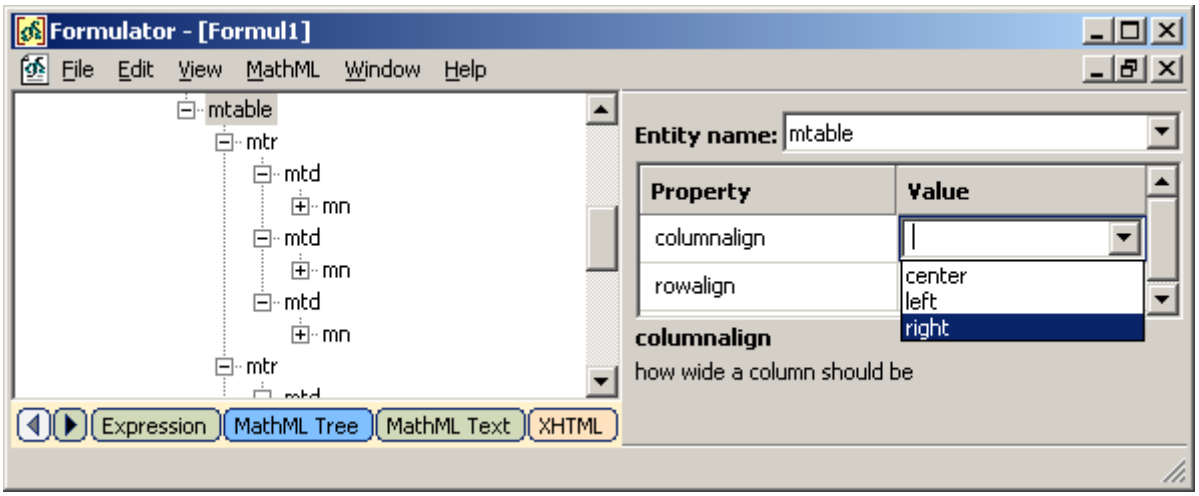
Click on the right side of the document to give it focus; then press the "Insert" button on the "Property-Value" pane. This will insert a new empty line for the attribute.



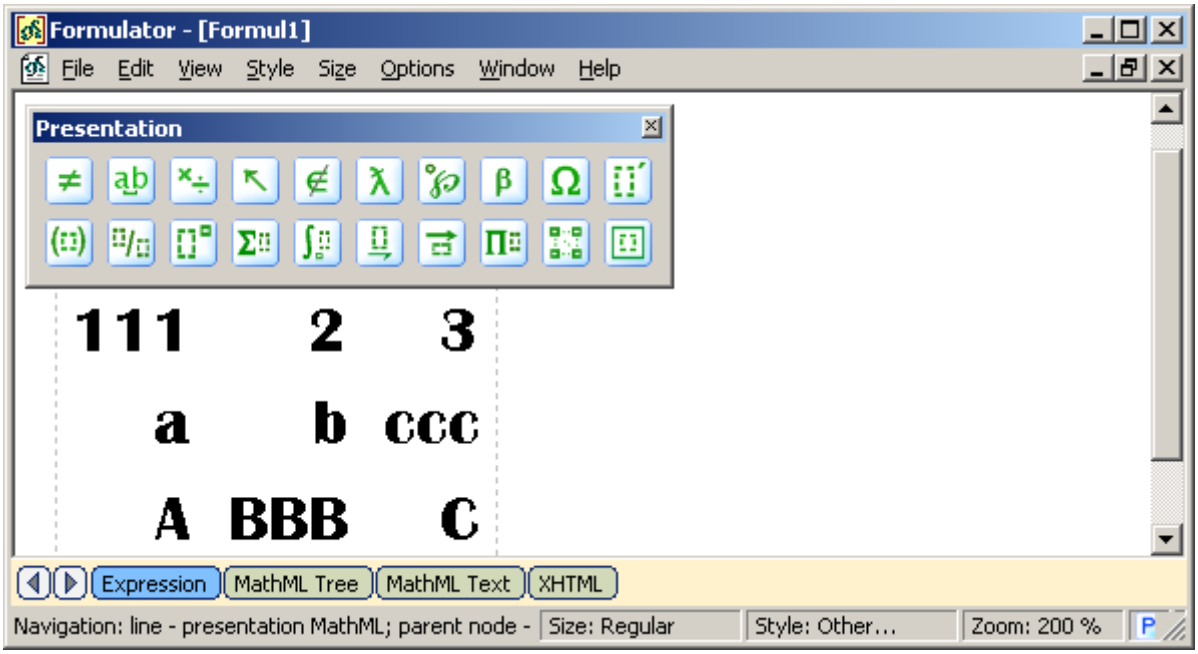
3. Click on the blank field of newly added attribute and use the drop list to get the list of all predefined attributes which are proper for the "mtable" element.



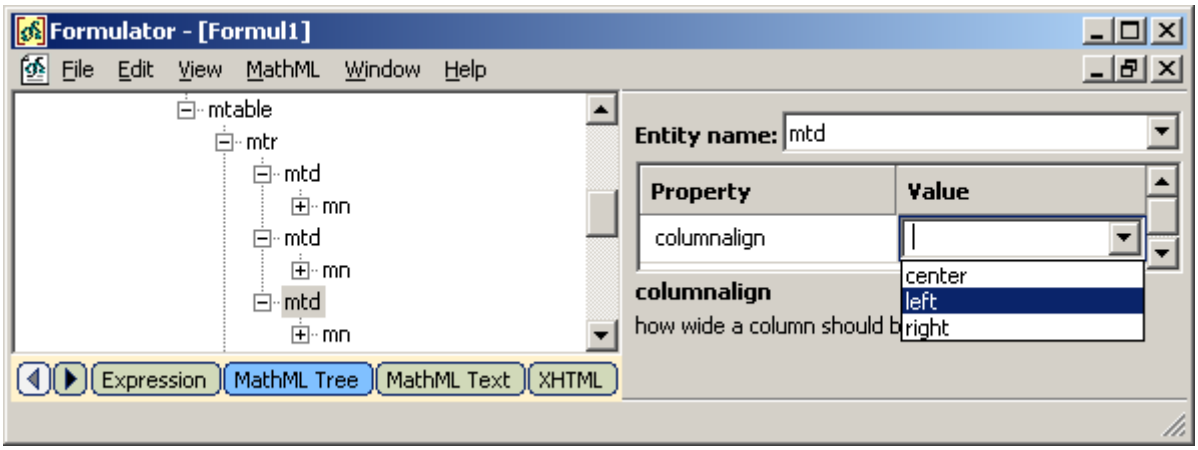
4. Select the "columnalign" attribute in the list; then place cursor to the "Value" column of the "Property-Value" pane and use the drop list to get the list of all predefined values which are proper for the "columnalign" attribute. Change the value of the attribute to the value of "right".

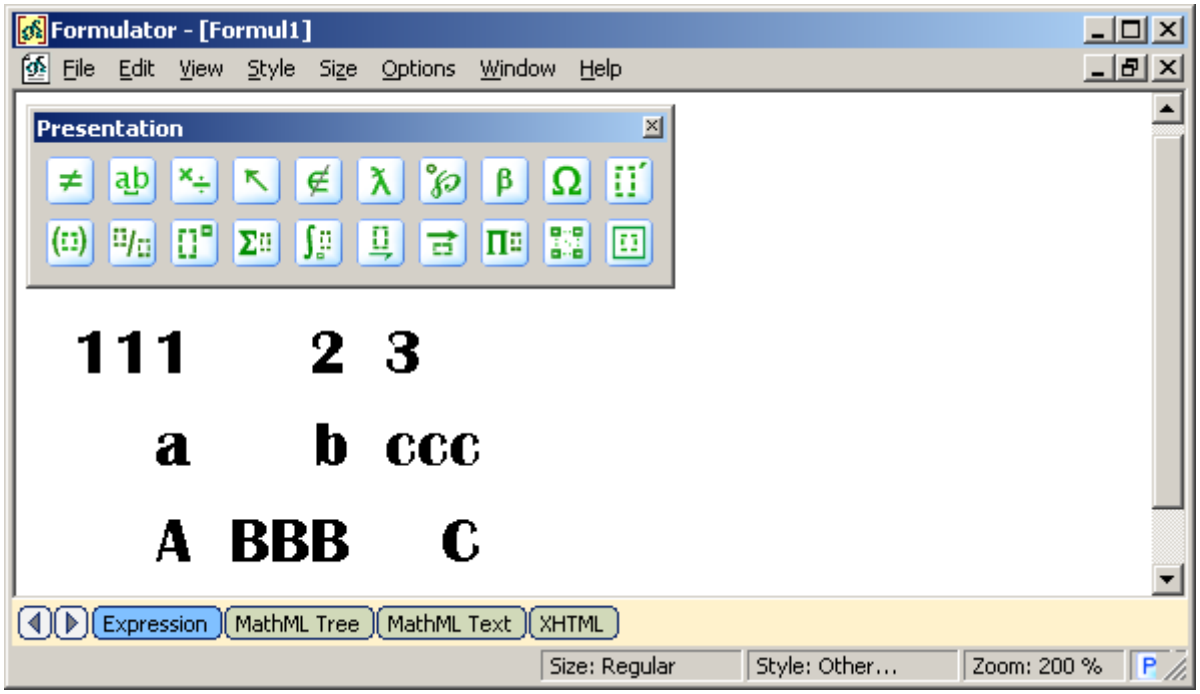


5. See how alignment of the whole table is changed.

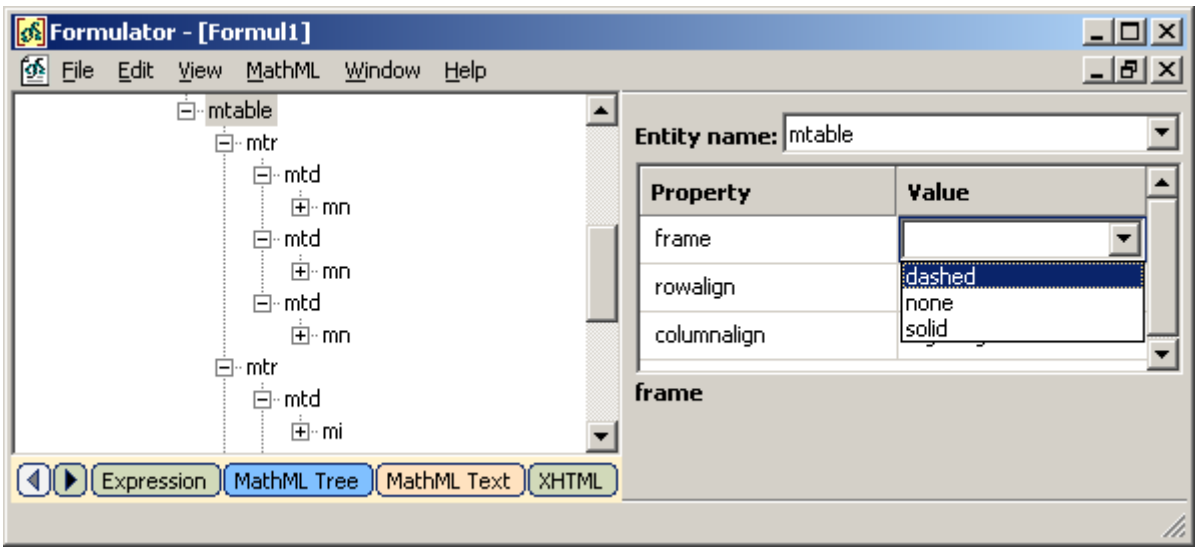


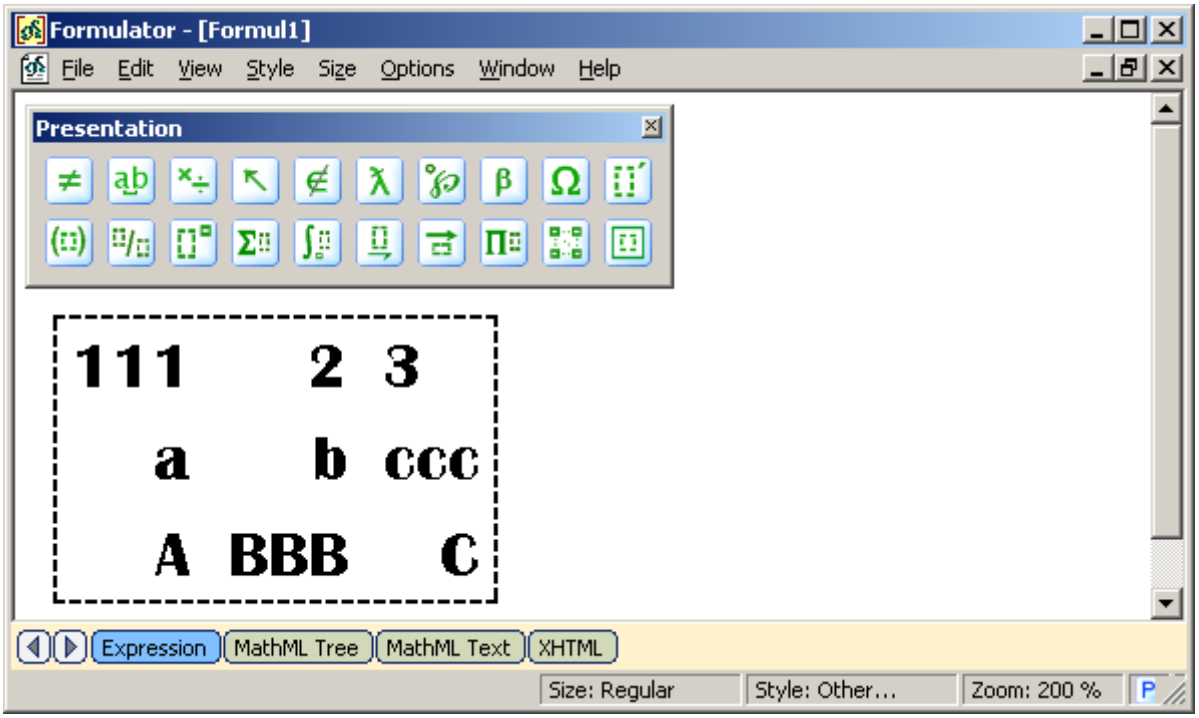
6. Using similar operations change alignment individually for the cell (1, 3) of the table.

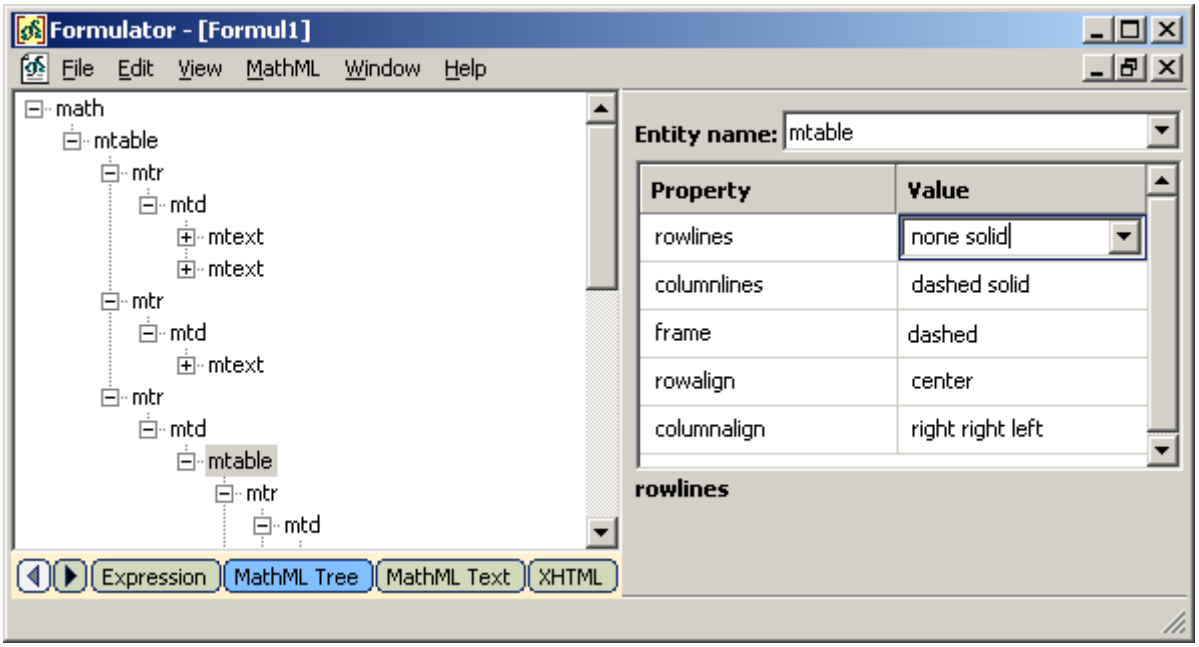


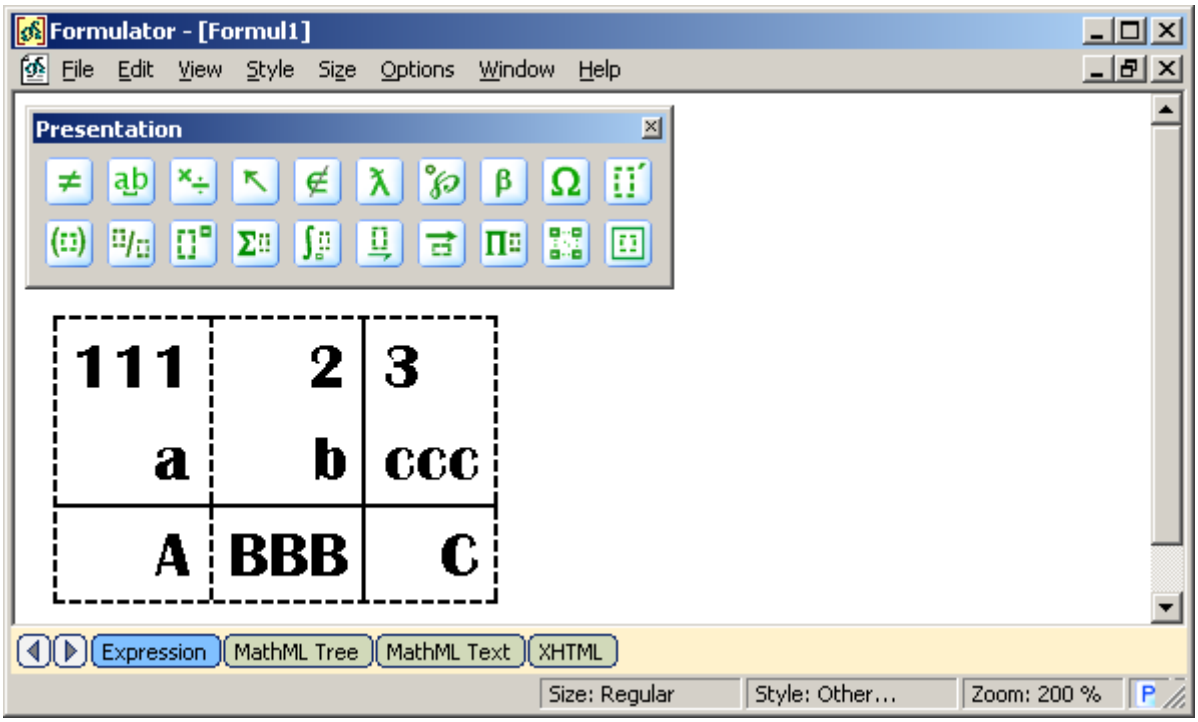


7. Switch to "MathML Tree" to change the frame of the table; switch back to see the results.




Formulator - [Formul1]
File Edit View Style Size Options Window Help
Presentation
111 2 3
a b ccc
A BBB C
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Other...
Zoom: 200 %

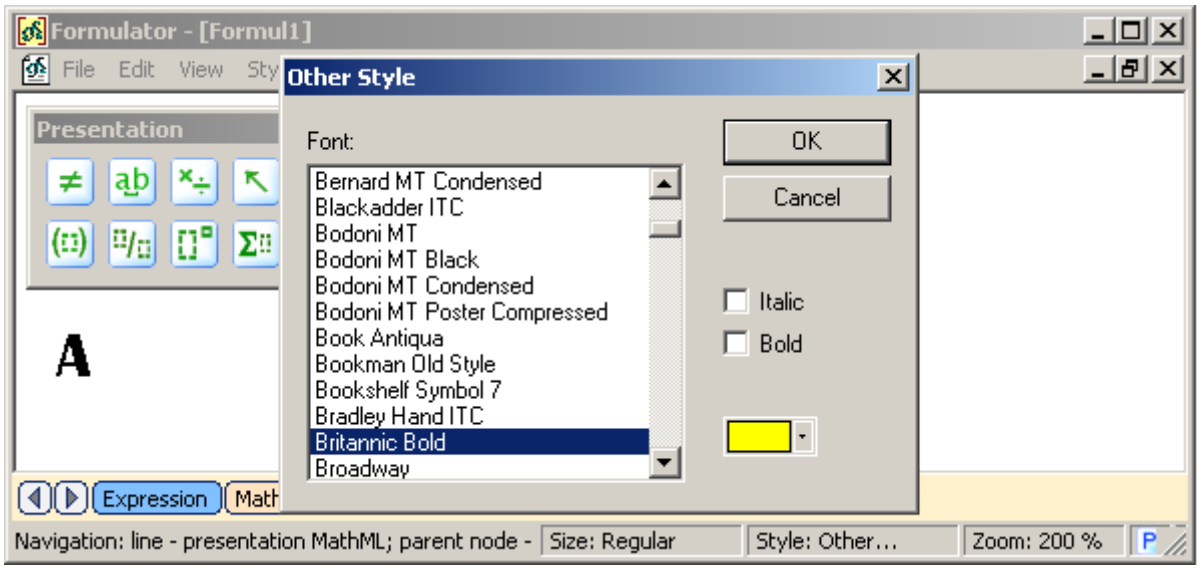

Formulator - [Formul1]
File Edit View MathML Window Help
math
mtable
mtr
mtd
mtext
mtext
mtr
mtd
mtext
mtr
mtd
mtable
mtr
mtd
Entity name: mtable
Property Value
rowlines none solid
columnlines dashed solid
frame dashed
rowalign center
columnalign right right left
rowlines
Expression MathML Tree MathML Text XHTML

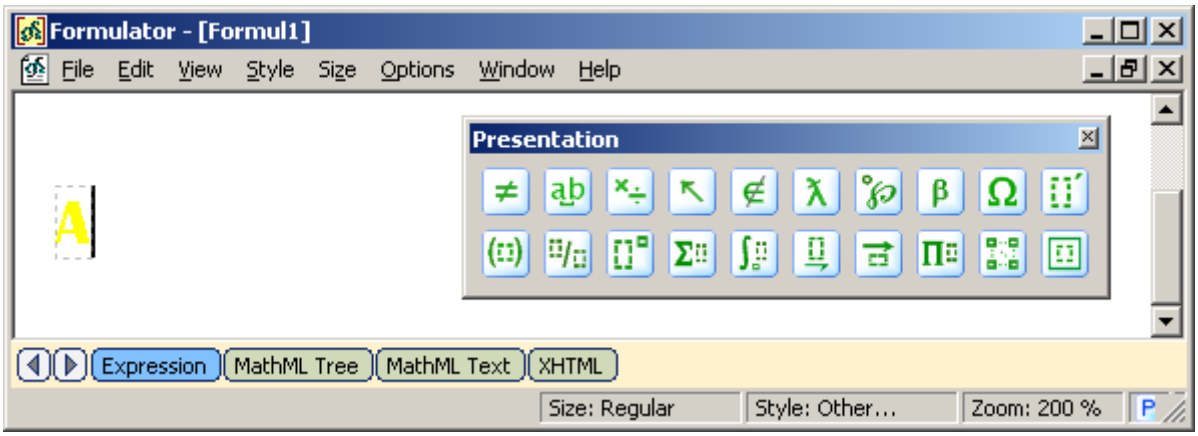


Style Change

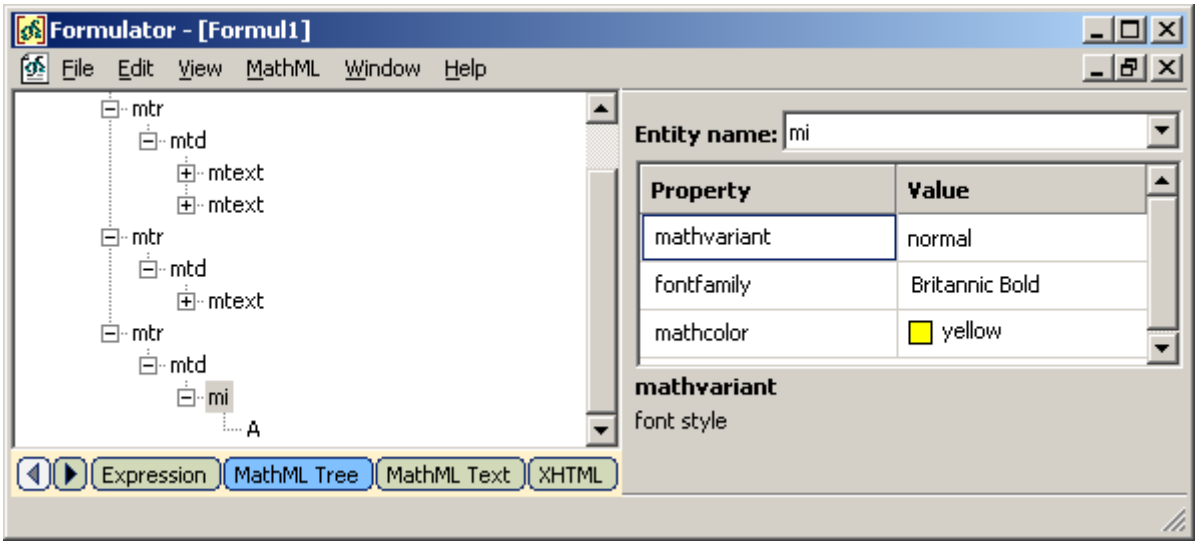
There are two ways to deal with "mstyle" element and style attributes in Formulator MathML Weaver.

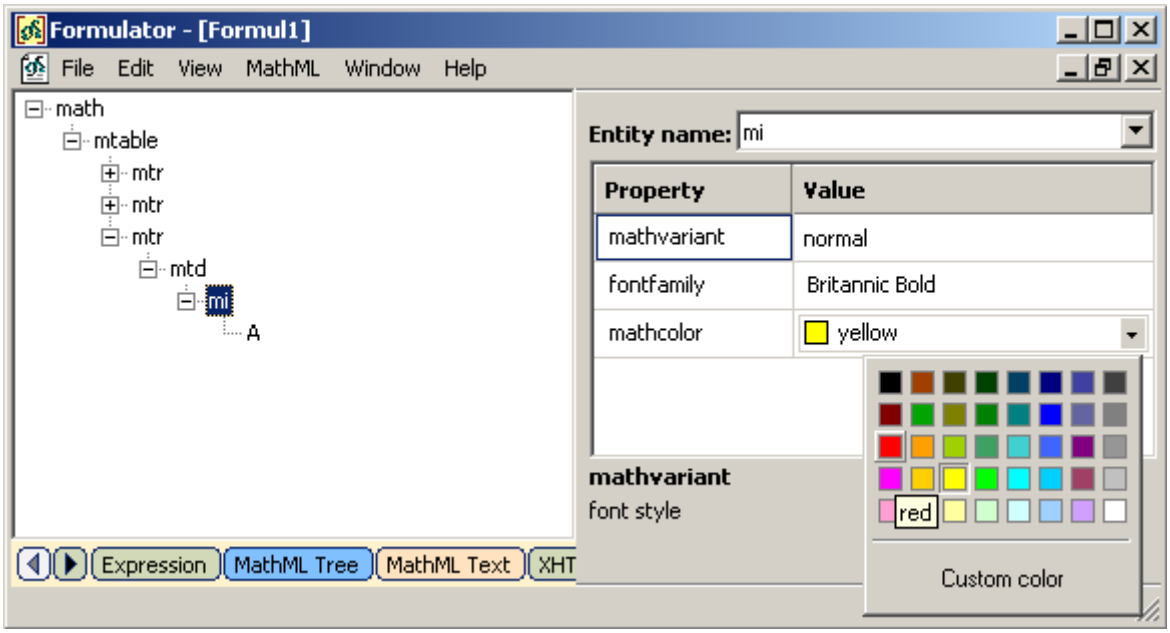
The first way is implicitly changing of style of formula elements by using menu Style and Size.

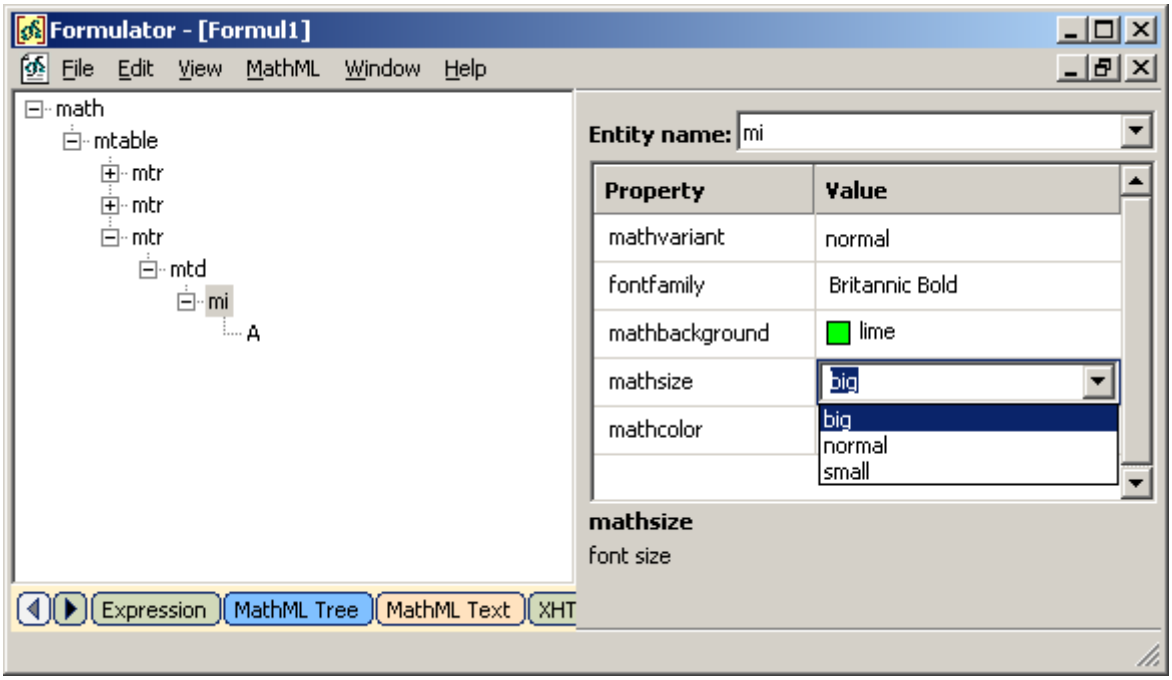


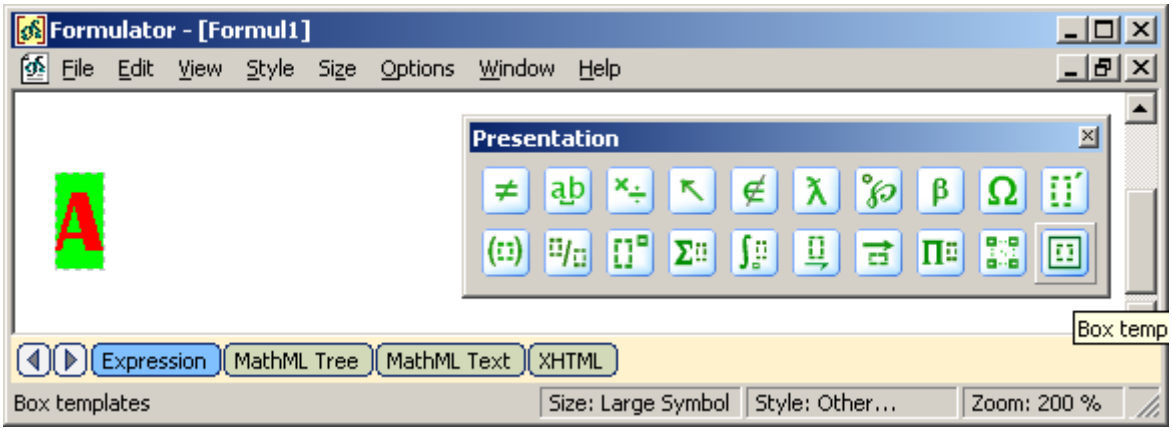


Then we can switch to the "MathML Tree" page and proceed with editing via WYSIWYG-style actions.



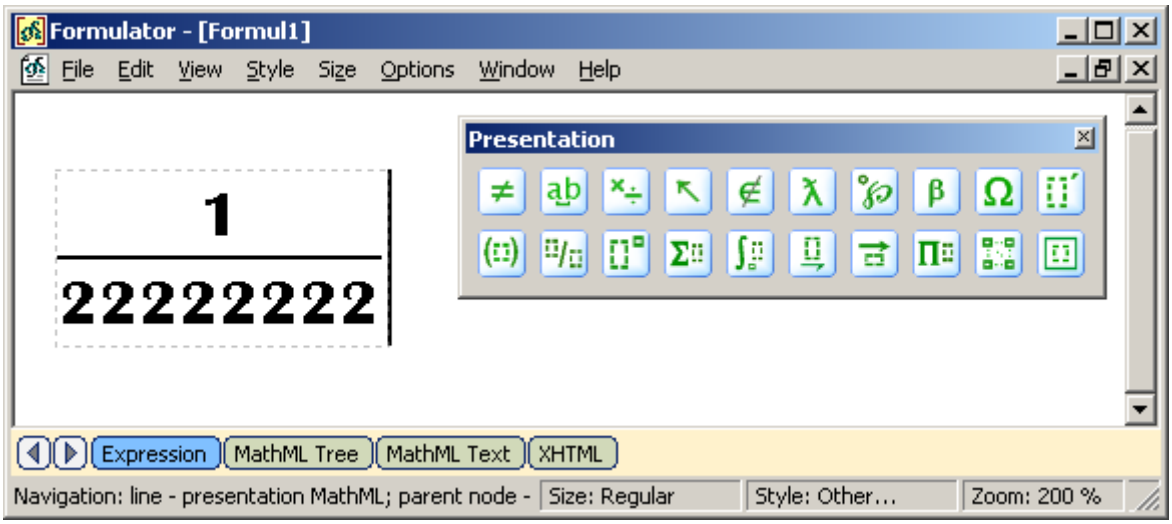


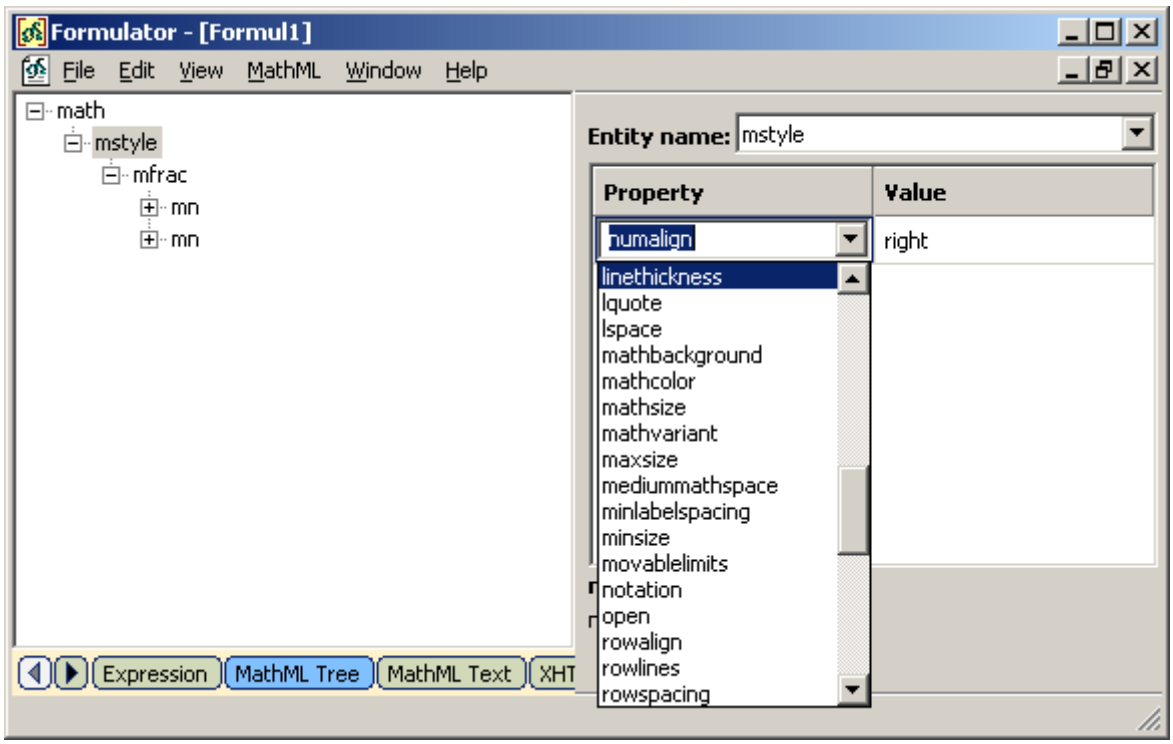


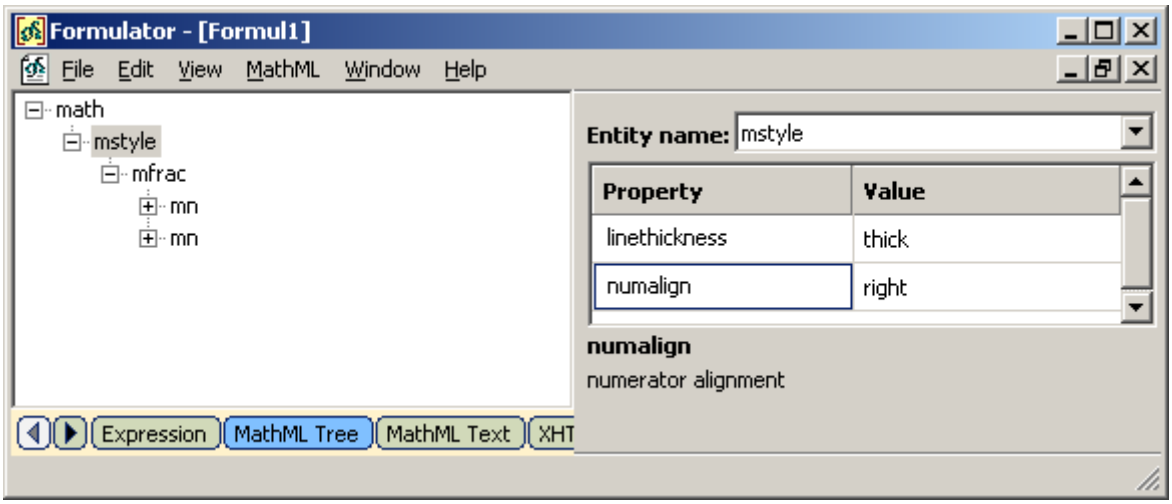


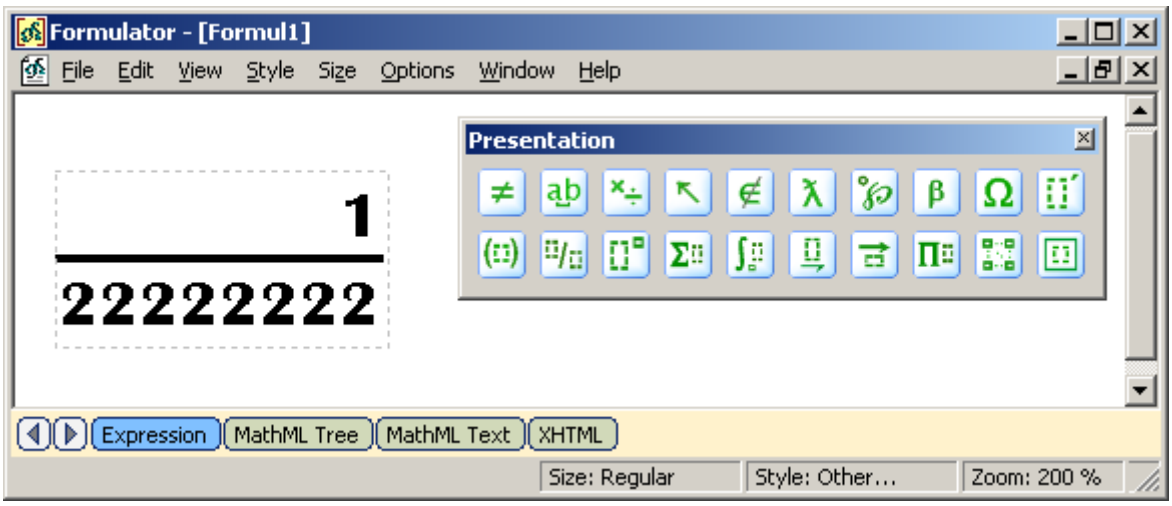
The second way of style changing is to insert the "mstyle" element on the "MathML Tree" page and to edit it using operations with nodes of the tree. Note that the "mstyle" element is inserted as an empty invisible "mrow" element that carries its own set of attributes.

1. The next example shows how to change alignment of the fraction elements using attributes of the "mstyle" element.

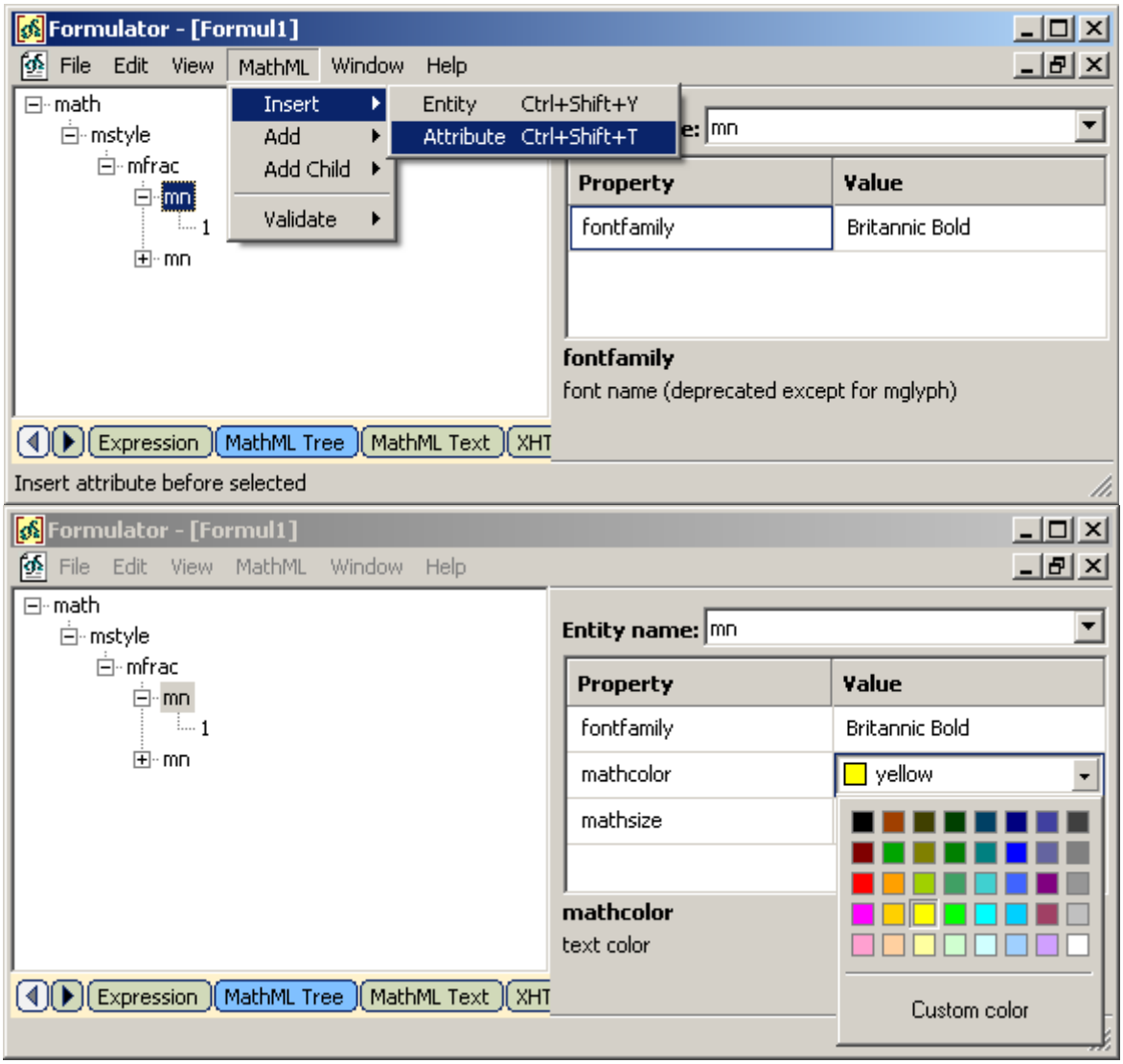

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Presentation
1
22222222
Expression MathML Tree MathML Text XHTML
Navigation: line - presentation MathML; parent node -
Size: Regular
Style: Other...
Zoom: 200 %

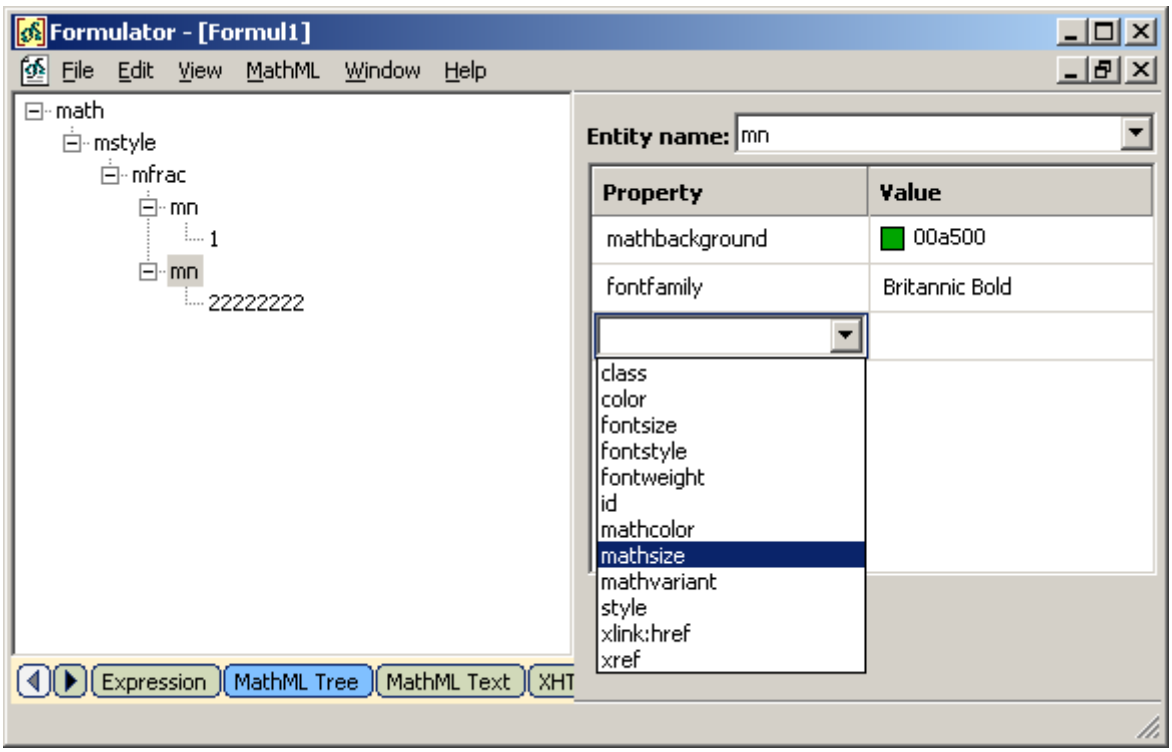

Formulator - [Formul1]
File Edit View MathML Window Help
math
mstyle
mfrac
mn
mn
Entity name: mstyle
Property
Value
numalign
right
linethickness
lquote
lspace
mathbackground
mathcolor
mathsize
mathvariant
maxsize
mediummathspace
minlabelspacing
minsize
movablelimits
notation
open
rowalign
rowlines
rowspacing
Expression MathML Tree MathML Text

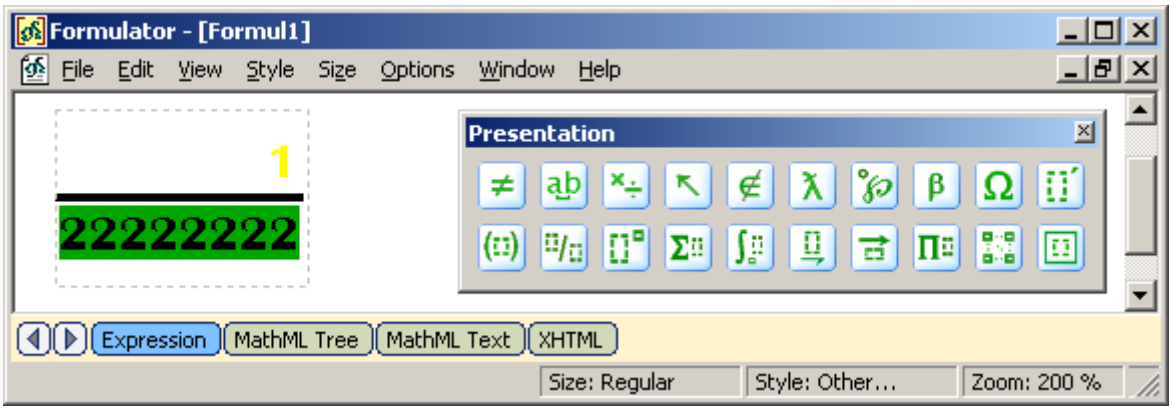

Formulator - [Formul1]
File Edit View MathML Window Help
math
mstyle
mfrac
mn
mn
Entity name: mstyle
Property
Value
linethickness
thick
numalign
right
numalign
numerator alignment
Expression MathML Tree MathML Text



2. The next example shows how to change appearance of token elements by editing attributes, related to a font.



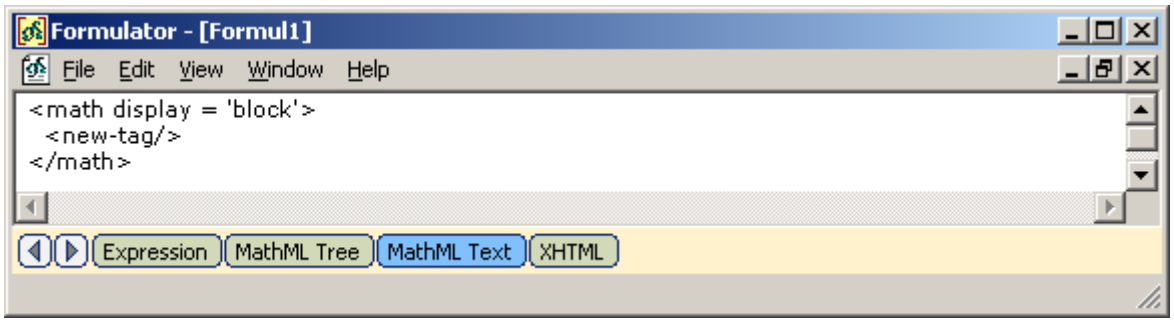


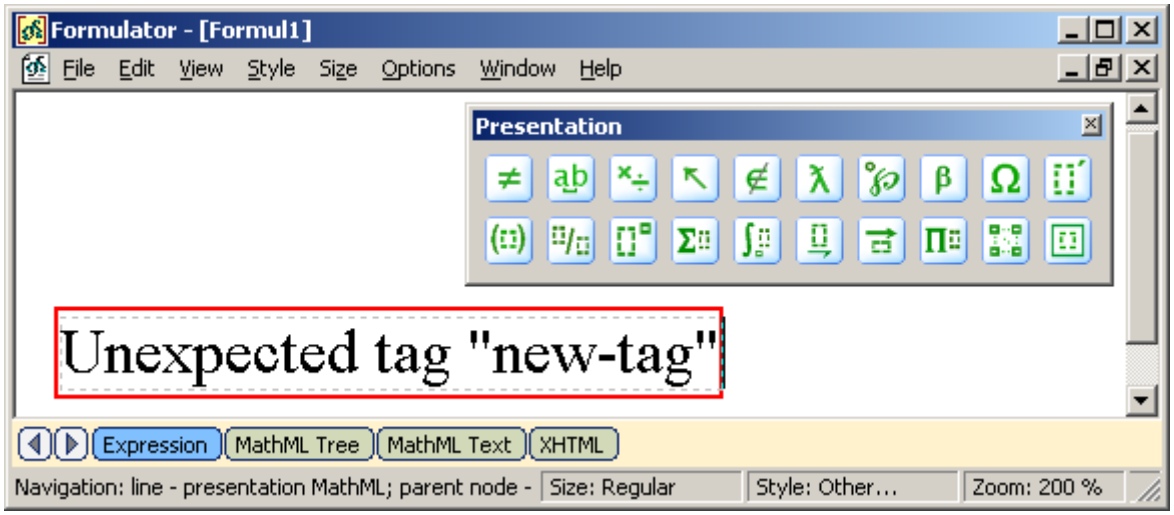


Error Message

A user can enter a new tag name on the "MathML Text" page, since it is just a text editor addition to MathML Weaver. This case should be treated as a user error, because the editor don't know how to deal with unknown elements.

In order to get user know about errors Formulator MathML Weaver uses the "merror" element. In the next example a user enters unknown tag name "new-tag" and gets a message about error in MathML 2.0 notation.




Formulator - [Formul1]
File Edit View Style Size Options Window Help
Presentation
Unexpected tag "new-tag"
Expression
MathML Tree
MathML Text
XHTML
Navigation: line - presentation MathML; parent node -
Size: Regular
Style: Other...
Zoom: 200 %