FORMULATOR v3.8

MathML Suite

Mastering MathML
Content Markup

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

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

Mastering MathML Content Markup

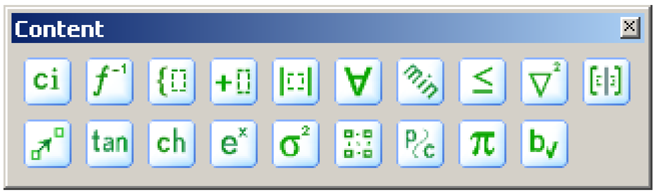
Presentation elements describe visual two-dimensional hierarchical forms and thus give more or less precise instructions how to render and how edit mathematical constructs. Content Markup closely follows the semantic structure of the mathematical objects. This is quite the challenging approach to coding mathematics for such a WYSIWYG-style mathematical editor, as Formulator 3.7 MathML Weaver.

As an answer to this problem MathML Weaver implements a technology of internal wrapping of Content Markup expressions in Presentation Markup nodes. The main tool for this is an empty frame element that encodes a new hierarchical relations inside of Content Markup expression and carries all the specific information about final names and values of tags and attributes. This allows to edit Content Markup expressions in WYSIWYG-style, but at the same to establish explicit connections between mathematical structures and their mathematical meanings.

Please note that, as a rule, a user should not care about wrapping of Content Markup expressions in Presentation Markup nodes. From the point of view of end-user, there is just a way in MathML Weaver to insert and to edit visually a semantic kind of MathML notation . The only reason why we raise this question in the manual is that bearing in mind of such an peculiarity of the Content Markup editing implementation can be useful for understanding of the behavior of MathML Weaver in some complicated cases.

General Principles of Editing and Navigation

MathML Weaver proposes two ways to create Content Markup elements. The first is most general, since it supposes to make use of mathematical toolbars:

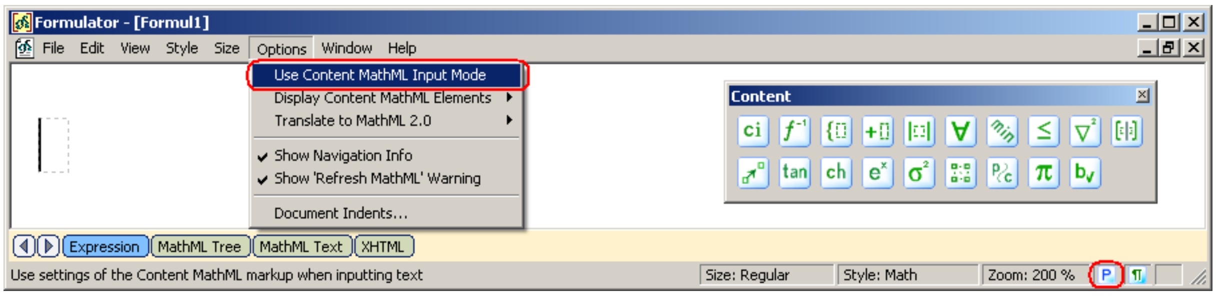


A group of Content mathematical templates comprises:

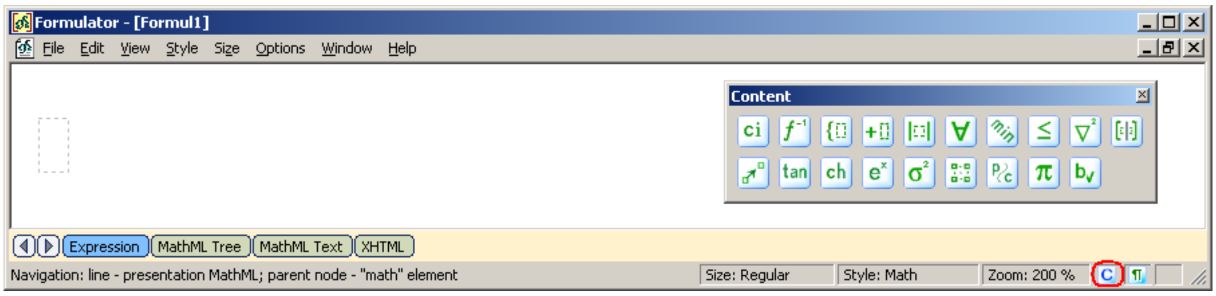
- token elements
- basic content elements
- piecewise declaration templates
- arithmetic operators
- algebra operators
- logic operators
- maximum and minimum templates
- relations templates
- calculus and vector calculus templates
- theory of sets templates
- sequences and series templates
- common trigonometric functions
- common hyperbolic functions
- common exponential functions
- statistics templates

- linear algebra templates
- semantics templates
- constant and symbol elements
- qualifier elements templates

The second way to insert Content Markup elements is connected with an option to switch input between Content and Presentation MathML input modes (the last one is used by default):


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Use Content MathML Input Mode
Display Content MathML Elements
Translate to MathML 2.0
Show Navigation Info
Show 'Refresh MathML' Warning
Document Indents...
Content
Expression MathML Tree MathML Text XHTML
Size: Regular Style: Math Zoom: 200 % P

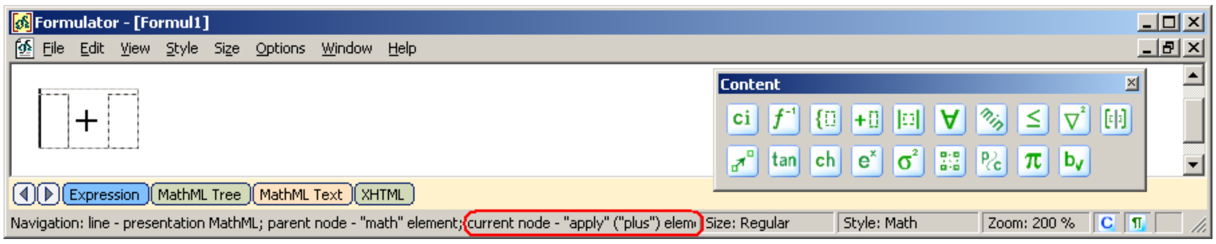
Use settings of the Content MathML markup when inputting text


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular Style: Math Zoom: 200 % C

Navigation: line - presentation MathML; parent node - "math" element

Selecting Content MathML Input Mode leads to inserting of Content MathML mathematical templates when a user presses a sign of the corresponding operation. E.g., pressing '+' in the Content MathML Input Mode inserts a mathematical template ▓+▓ for the <apply> element with the operator element <plus/>. The next example shows how to work with this mode in more details.

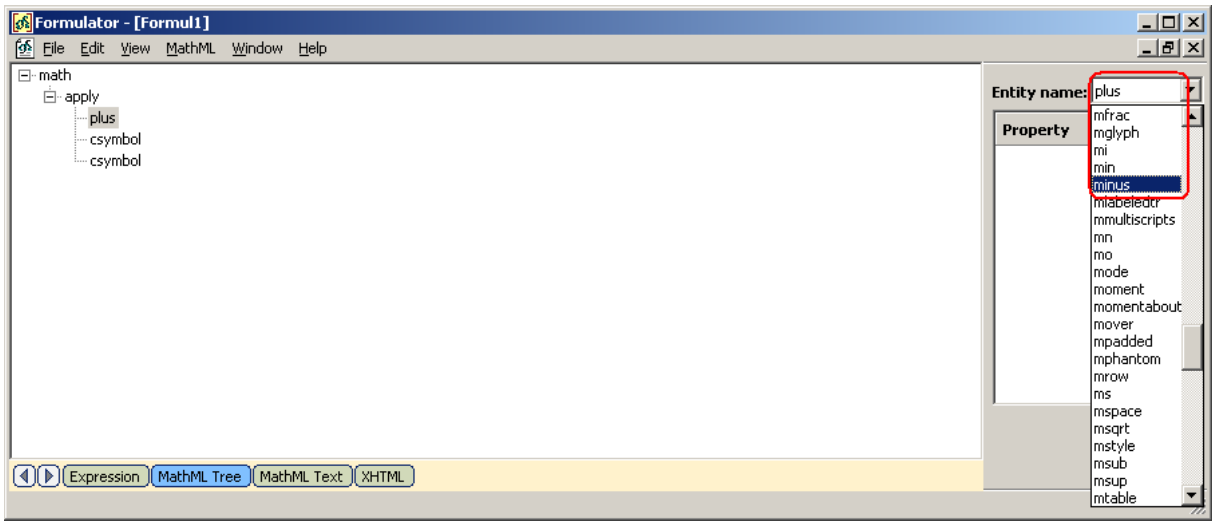
1. After switching to the Content MathML Input Mode type '+' sign (make sure that Status Icon in the right bottom corner of MathML Weaver indicates C). We now see the <apply> element inserted. Then we can use the "Show Nesting" command and navigation arrows to investigate how this Content Markup element is composed.

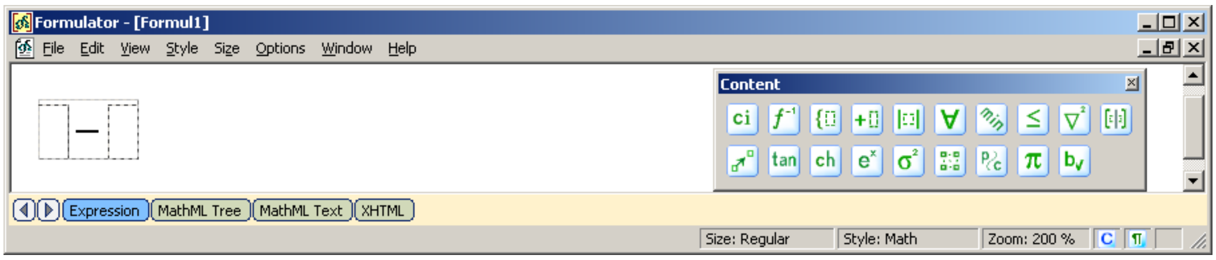

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular Style: Math Zoom: 200 % C

Navigation: line - presentation MathML; parent node - "math" element; current node - "apply" ("plus") elem

A grey rectangle around the ▓+▓ form and tips in the status bar suggests that in the document there is one frame node, marked as an "apply" element and internally encoded as an additional nesting "mrow" element.

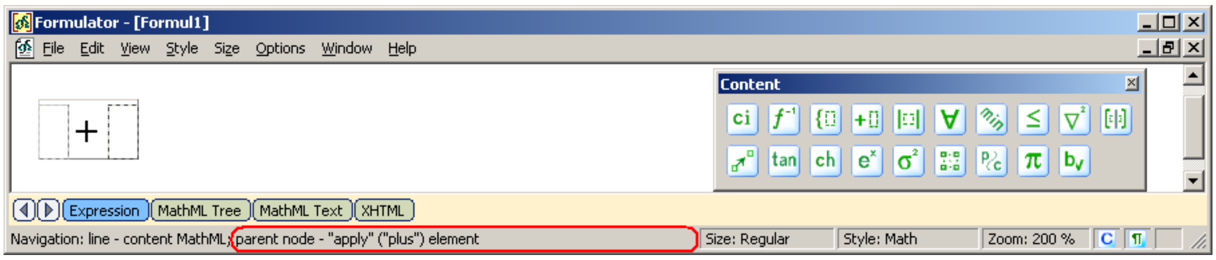
After pressing the Right arrow we consequently get to the input slots of this "apply" form. '+' sign cannot be edited on this stage, since it is an integral part of the expression. For the sake of flexibility the type of operation still can be easily changed using the "MathML Tree" page:



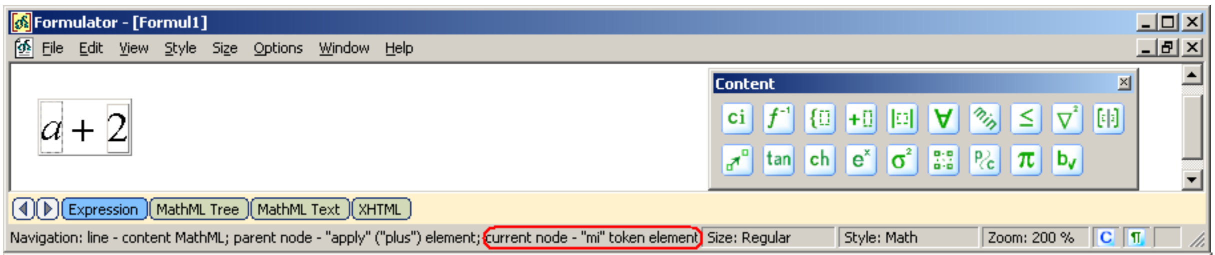

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

2. Now we can enter arguments of the just created "apply" element. The most important thing about input slots of this visual form is that their kinds can be automatically detected by MathML Weaver after a user types some text or utilizes Formulator's mathematical toolbars.

Initially, these input slots don't have any predefined kind (associated Content or Presentation Markup element). It can be easily discovered if we look on the navigation information bar; there is a record for the parent node, but the current node record is undefined yet.

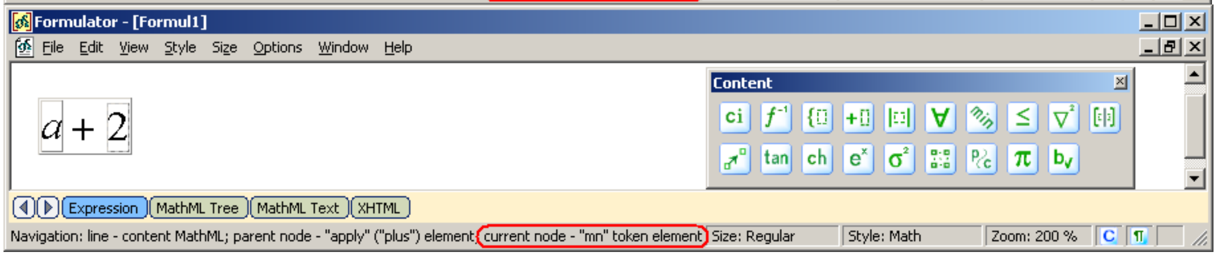

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Navigation: line - content MathML; parent node - "apply" ("plus") element
Size: Regular
Style: Math
Zoom: 200 %

Type character 'a' in the first slot, press the Right arrow and type '2' in the second slot:


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci tan ch π
Expression MathML Tree MathML Text XHTML

$a + 2$

Navigation: line - content MathML; parent node - "apply" ("plus") element; current node - "mi" token element Size: Regular Style: Math Zoom: 200 %

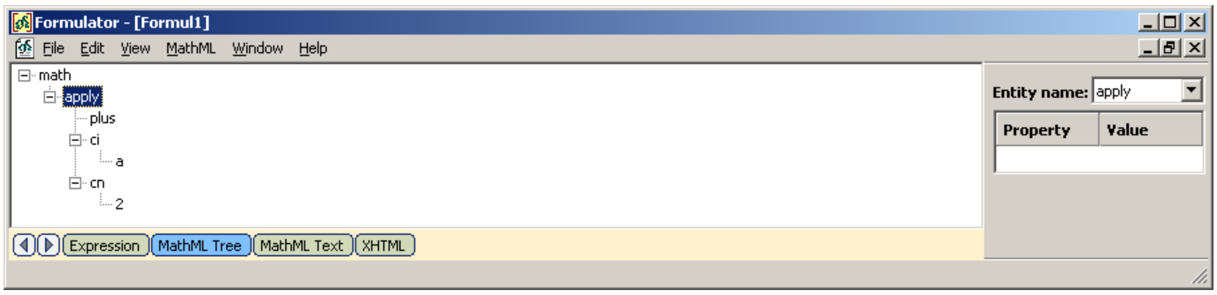

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci tan ch π
Expression MathML Tree MathML Text XHTML

$a + 2$

Navigation: line - content MathML; parent node - "apply" ("plus") element; current node - "mn" token element Size: Regular Style: Math Zoom: 200 %

Note that on this stage 'a' and '2' are still marked as presentation elements ("mi", "mn"). They become Content elements only *after* they will be converted from the internal document representation to MathML 2.0 text.

Switch to the "MathML Tree" page and make certain that the "apply" element is created correctly.

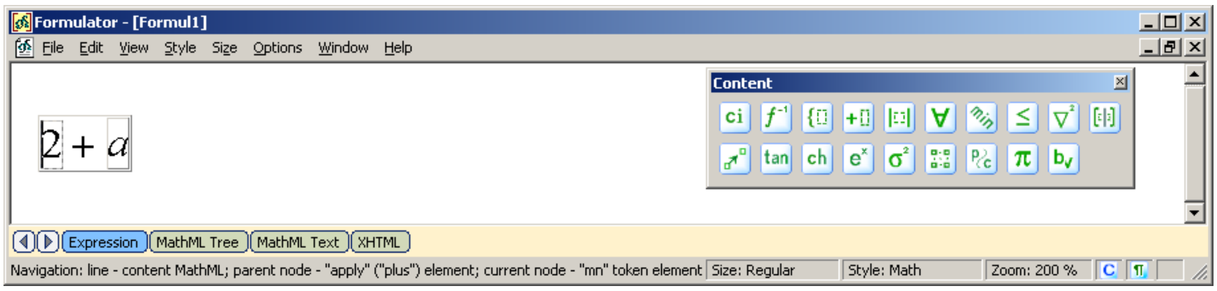

Formulator - [Formul1]
File Edit View MathML Window Help
Expression MathML Tree MathML Text XHTML

```
math
  apply
    plus
    ci
      a
    cn
      2
```

Entity name: apply

Property	Value

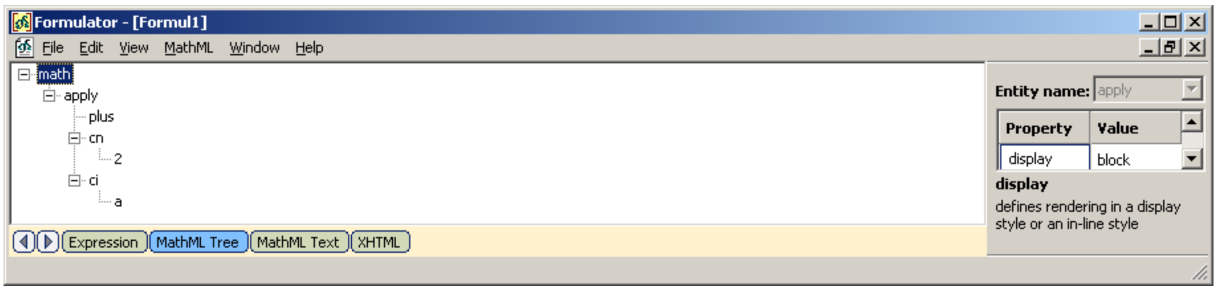
Now switch back to the "Expression" page and exchange 'a' and '2' in their input slots.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci tan ch π
Expression MathML Tree MathML Text XHTML

$2 + a$

Navigation: line - content MathML; parent node - "apply" ("plus") element; current node - "mn" token element Size: Regular Style: Math Zoom: 200 %

The “MathML Tree” page shows that input slots still work as automatically detected by their content areas, because the first slot now turns to be ”cn” element and the second slot became the ”ci” element.


Formulator - [Formul1]
File Edit View MathML Window Help
Expression MathML Tree MathML Text XHTML

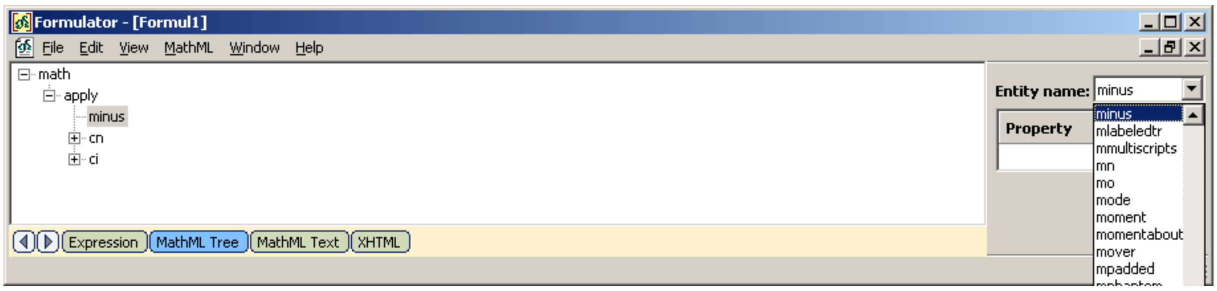
```
math
  apply
    plus
    cn
      2
    ci
      a
```

Entity name: apply

Property	Value
display	block

display
defines rendering in a display style or an in-line style

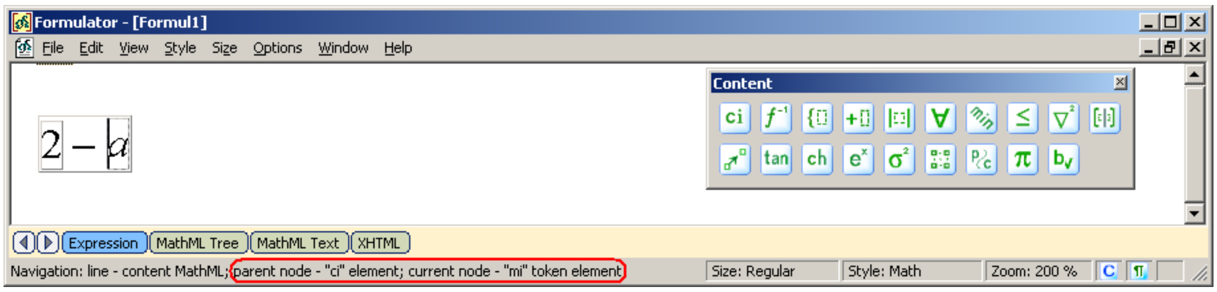
3. This smart behavior of the input slots of the “apply” form lasts for all the time of WYSIWYG-style editing, but stops when a user manually interferes in the editing by means of the “MathML Tree” or “MathML Text” page. To see this, switch to the “MathML Tree” and change type of the operator from <plus/> to <minus/>:


Formulator - [Formul1]
File Edit View MathML Window Help
Expression MathML Tree MathML Text XHTML
Property

```
math
  apply
    minus
    cn
    ci
```

Entity name: minus

minus
mlabeledtr
mmultiscripts
mn
mo
mode
moment
momentabout
mover
mpadded

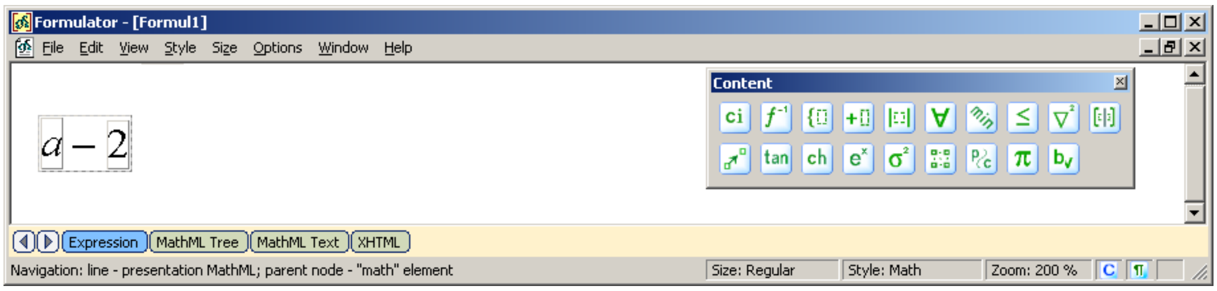

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci
tan
ch
Expression MathML Tree MathML Text XHTML

$2 - a$

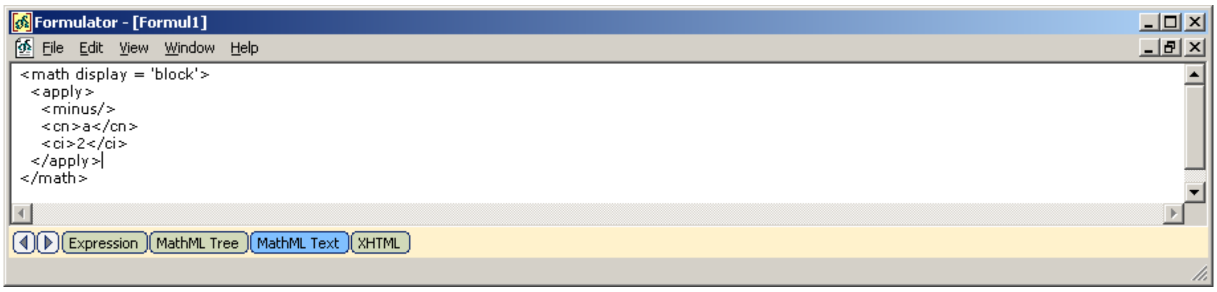
Navigation: line - content MathML; parent node - "ci" element; current node - "mi" token element | Size: Regular | Style: Math | Zoom: 200 %

At first glance nothing changed, but take a look to the navigation information bar (red circled area). Now We have a rigid Content Markup structure, composed of the “apply” element with a leaf of the “ci” element. The internal presentation of the “ci” element is formed of the “mi” element, but it is of less importance, since in the MathML 2.0 text there will be no internal presentation remains.

To make sure that a new structure of the expression is fixed try to repeat our trick with exchanging ‘a’ and ‘2’ in their input slots.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular Style: Math Zoom: 200 %

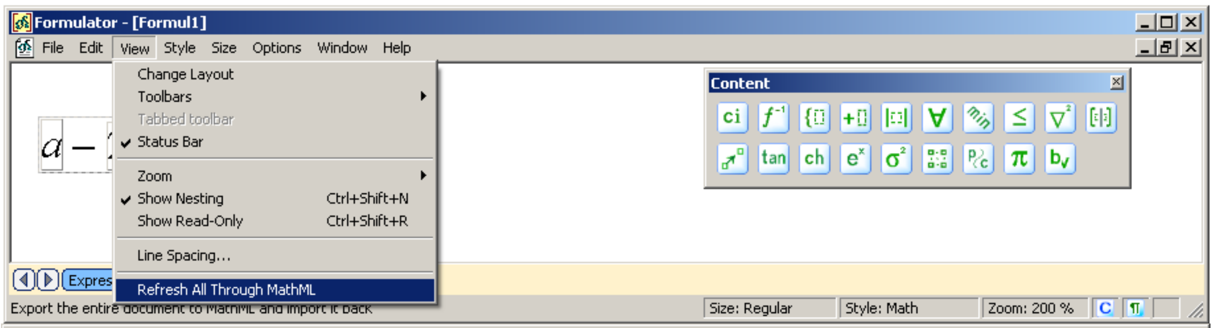
Navigation: line - presentation MathML; parent node - "math" element


Formulator - [Formul1]
File Edit View Window Help
Expression MathML Tree MathML Text XHTML

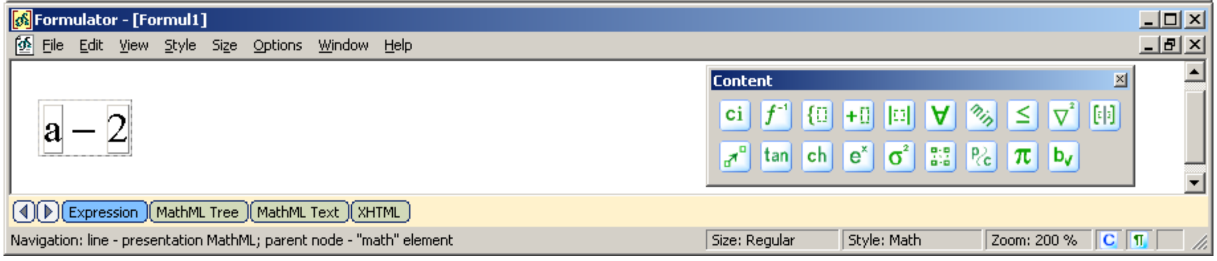
```
<math display = 'block'>
  <apply>
    <minus/>
    <cn>a</cn>
    <ci>2</ci>
  </apply>
</math>
```

Now ‘a’ is considered as a number token element “cn” and ‘2’ is considered as an identifier element “ci”, because in the *rigid structured document* input slots have already known their kind of Content MathML elements and no automatically detection was provided.

In this example we show another typical feature of the way that MathML Weaver deals with Content Markup. Since ‘a’ is situated in the “cn” element, it should not be rendered as italic. But MathML Weaver doesn’t interfere into a user’s work in such a case, but proposes instead a short way to bring the formula’s image to conformity with its semantics. Choose the “Refresh All Through MathML” command from the View menu and see how the entire look of the mathematical formula is changed.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Change Layout
Toolbars
Tabbed toolbar
Status Bar
Zoom
Show Nesting Ctrl+Shift+N
Show Read-Only Ctrl+Shift+R
Line Spacing...
Refresh All Through MathML
Content
Size: Regular Style: Math Zoom: 200 %

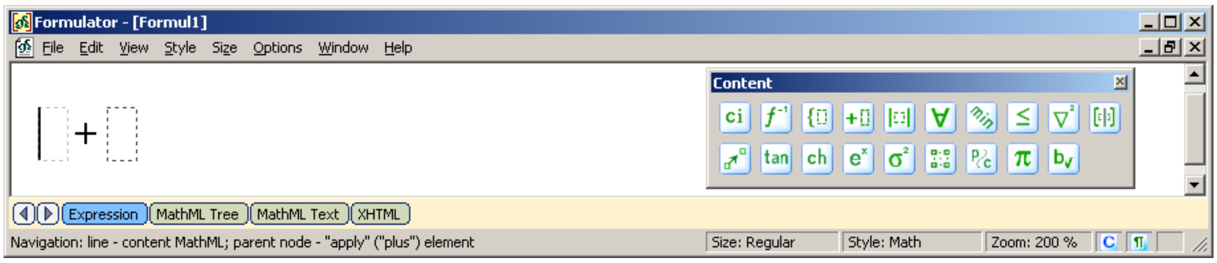
Export the entire document to MathML and import it back


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular Style: Math Zoom: 200 %

Navigation: line - presentation MathML; parent node - "math" element

3. Input slots of the created "apply" form can be filled as well with more complicated expressions than tokens. See, for example, how to create mathematical expression with several arithmetical operations.

Press the '+' sign.

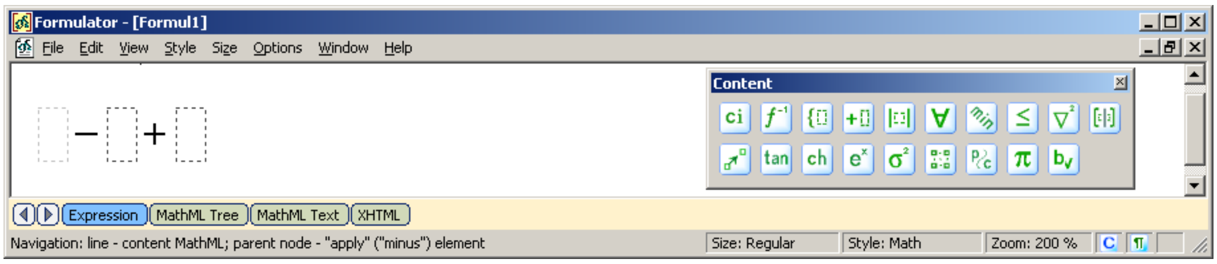

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

$\square + \square$

Navigation: line - content MathML; parent node - "apply" ("plus") element

Size: Regular Style: Math Zoom: 200 %

Press the '-' sign.

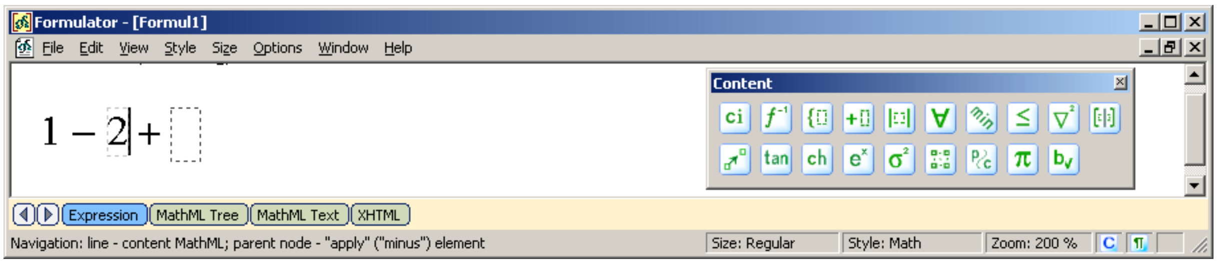

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

$\square - \square + \square$

Navigation: line - content MathML; parent node - "apply" ("minus") element

Size: Regular Style: Math Zoom: 200 %

Press '1', the Right arrow, '2'.

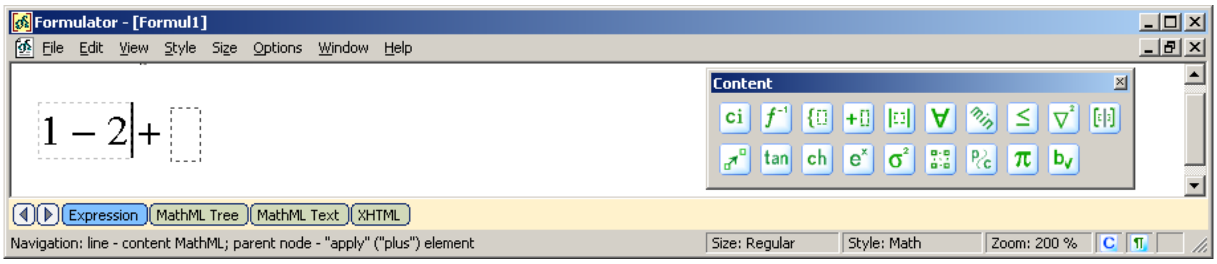

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

$1 - 2 + \square$

Navigation: line - content MathML; parent node - "apply" ("minus") element

Size: Regular Style: Math Zoom: 200 %

Press the Right arrow. Now be careful with editing actions, because dashed selection around 1 – 2 form indicates that we now on the hierarchical level of the "apply" element with <minus/> operator inside.

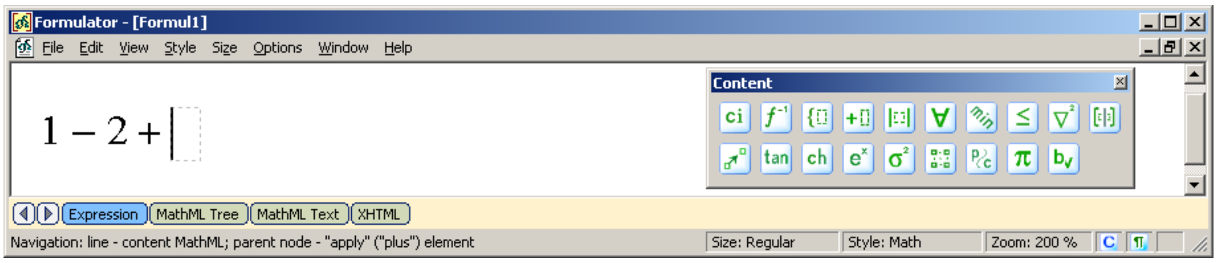

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

$1 - 2 + \square$

Navigation: line - content MathML; parent node - "apply" ("plus") element

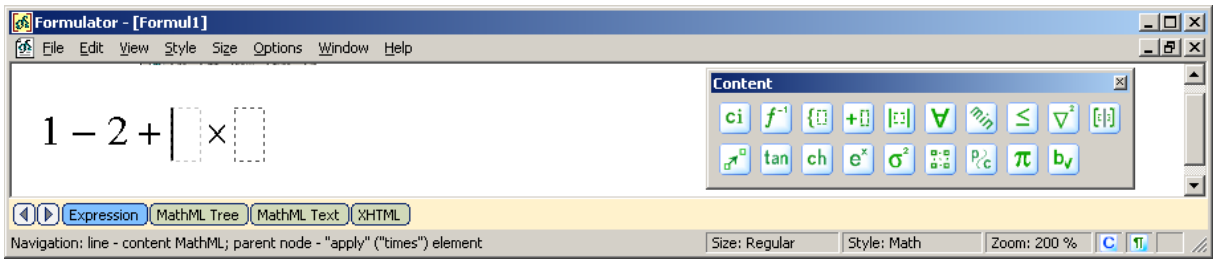
Size: Regular Style: Math Zoom: 200 %

To continue editing the formula we need to place cursor to an empty input slot on the right side of the formula, so, just press the Right arrow once more.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
1 − 2 +
Content
Expression MathML Tree MathML Text XHTML
Size: Regular Style: Math Zoom: 200 %

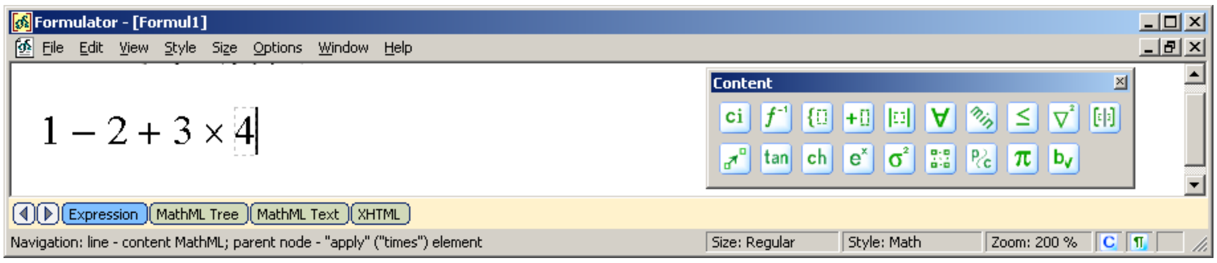
Navigation: line - content MathML; parent node - "apply" ("plus") element

Press ‘*’.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
1 − 2 + ×
Content
Expression MathML Tree MathML Text XHTML
Size: Regular Style: Math Zoom: 200 %

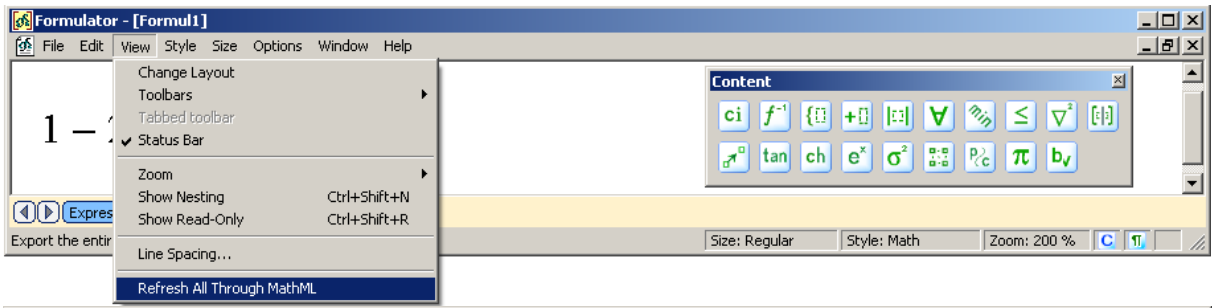
Navigation: line - content MathML; parent node - "apply" ("times") element

Press ‘3’, the Right arrow, ‘4’.

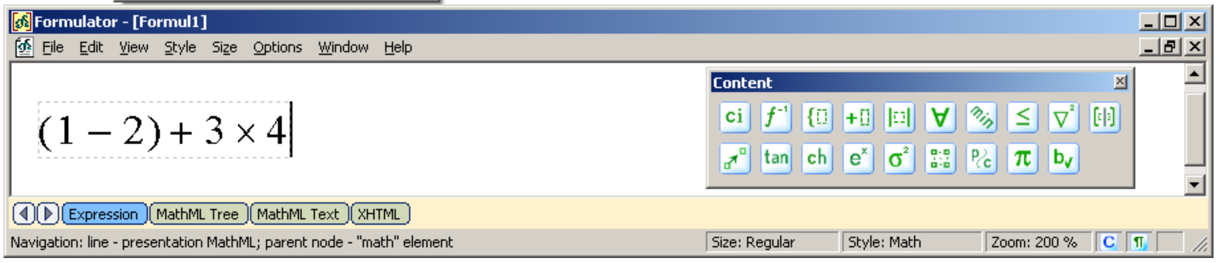

Formulator - [Formul1]
File Edit View Style Size Options Window Help
1 − 2 + 3 × 4
Content
Expression MathML Tree MathML Text XHTML
Size: Regular Style: Math Zoom: 200 %

Navigation: line - content MathML; parent node - "apply" ("times") element

The needed formula is created but lacks of proper appearance, because MathML Weaver has no chance yet to apply its precedence rules and to calculate how to draw this Content Markup formula. Choose the “Refresh All Through MathML” command and see how the image of the mathematical formula is changed.

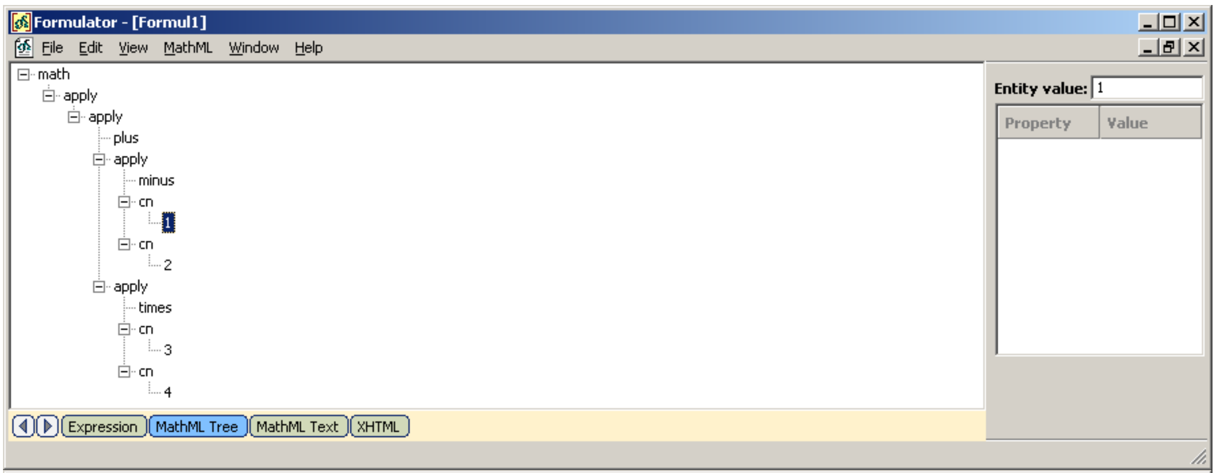

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Change Layout
Toolbars
Tabbed toolbar
✓ Status Bar
Zoom
Show Nesting Ctrl+Shift+N
Show Read-Only Ctrl+Shift+R
Line Spacing…
Refresh All Through MathML
Content
Size: Regular Style: Math Zoom: 200 %

Export the entir

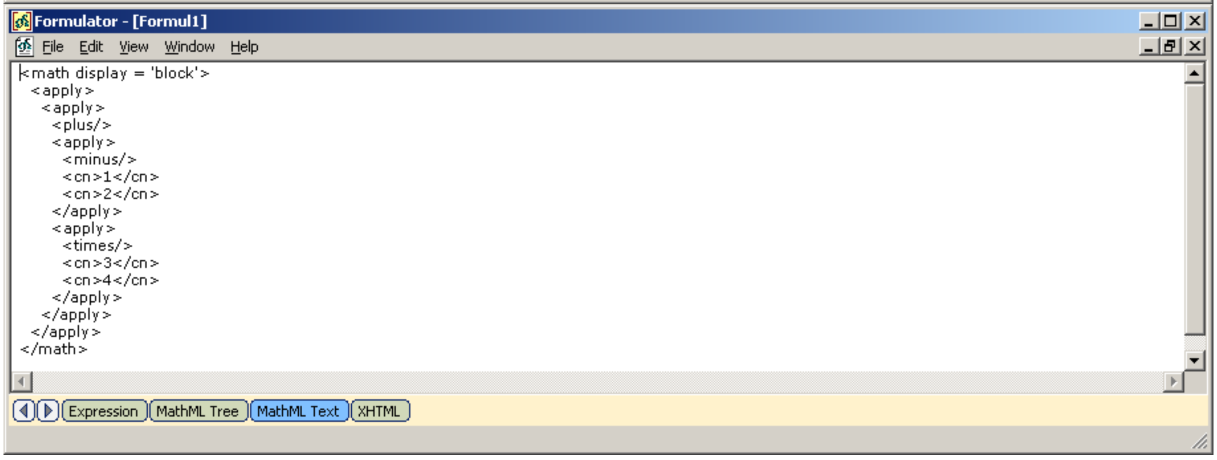

Formulator - [Formul1]
File Edit View Style Size Options Window Help
(1 − 2) + 3 × 4
Content
Expression MathML Tree MathML Text XHTML
Size: Regular Style: Math Zoom: 200 %

Navigation: line - presentation MathML; parent node - "math" element

After we have created a mathematical formula we can check how it is encoded in MathML 2.0 format using the “MathML Tree” or “MathML Text” pages.

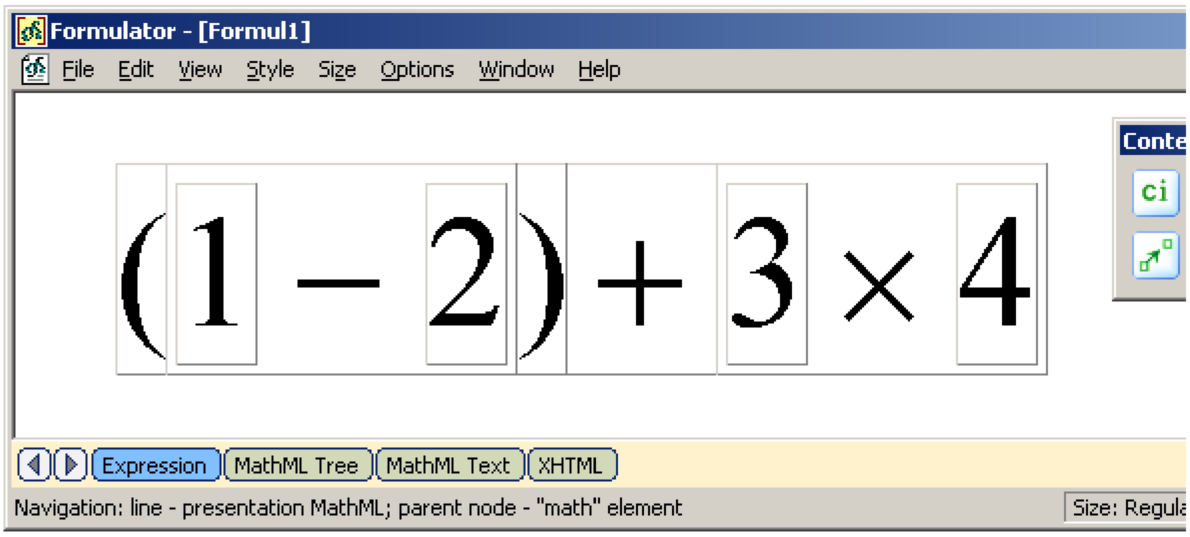

Formulator - [Formul1]
File Edit View MathML Window Help
Entity value: 1
Property Value
Expression MathML Tree MathML Text XHTML

```
math
  apply
    apply
      plus
      apply
        minus
        cn
          1
        cn
          2
      apply
        times
        cn
          3
        cn
          4
```


Formulator - [Formul1]
File Edit View Window Help
Expression MathML Tree MathML Text XHTML

```
<math display = 'block'>
 <apply>
  <apply>
   <plus/>
   <apply>
    <minus/>
    <cn>1</cn>
    <cn>2</cn>
   </apply>
   <apply>
    <times/>
    <cn>3</cn>
    <cn>4</cn>
   </apply>
  </apply>
 </apply>
</math>
```

4. Navigation through the newly created formula is not absolutely evident until we think about hierarchical structure of the Content Markup expression. Choose the "Show Nesting" command (by pressing Ctrl+Shift+N or from the "View" menu); then choose a new greater scale factor by pressing Ctrl+5 or from the "Zoom" submenu of the "View" menu.

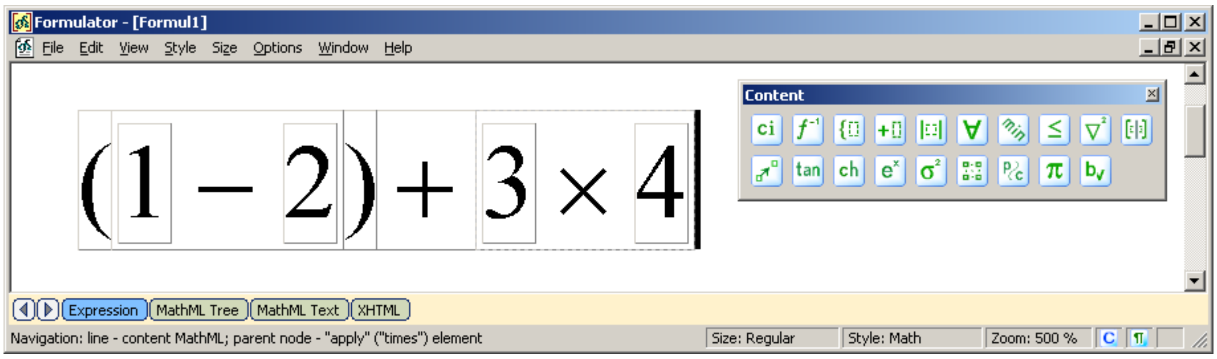

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Conte
ci
Expression MathML Tree MathML Text XHTML
Size: Regula

$(1-2)+3\times 4$

Navigation: line - presentation MathML; parent node - "math" element

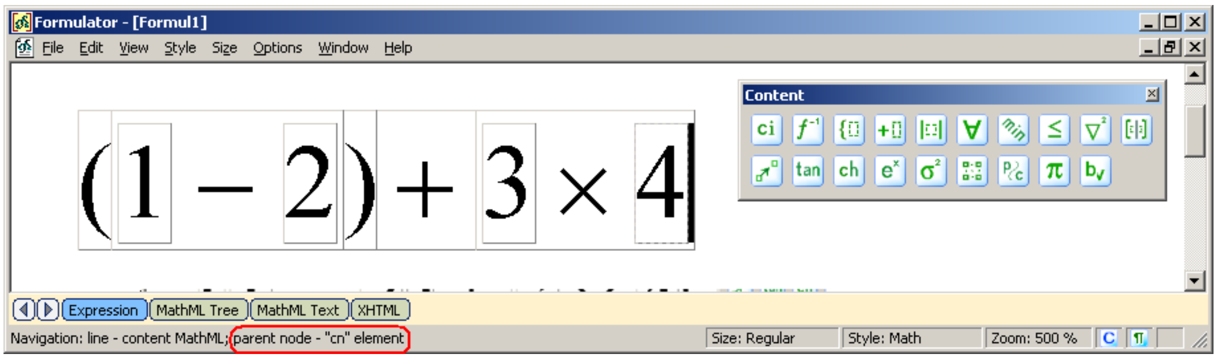
Now we can clearly see that the structure of the expression is not simple and we cannot expect the navigation to be as plain as in the Presentation Markup case. The higher level "apply" element has two child "apply" elements, and each of them in its turn has child leafs, encoded with token elements.

Press the mouse button on the right of the formula. This will place the cursor in the outermost right position available for editing.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

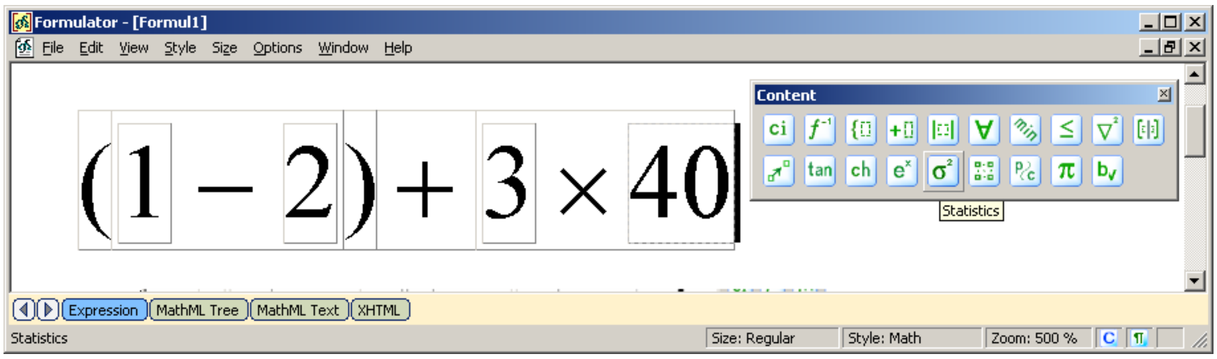
Navigation: line - content MathML; parent node - "apply" ("times") element | Size: Regular | Style: Math | Zoom: 500 %

Press the Left arrow. The cursor is now in a position for editing an input slot that currently has ‘4’ in it. See the navigation information bar to make sure of this. Note that cursor height is changed also according to the current inner level of hierarchy.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

Navigation: line - content MathML; parent node - "cn" element | Size: Regular | Style: Math | Zoom: 500 %

Press ‘0’ to edit the last argument of the expression.

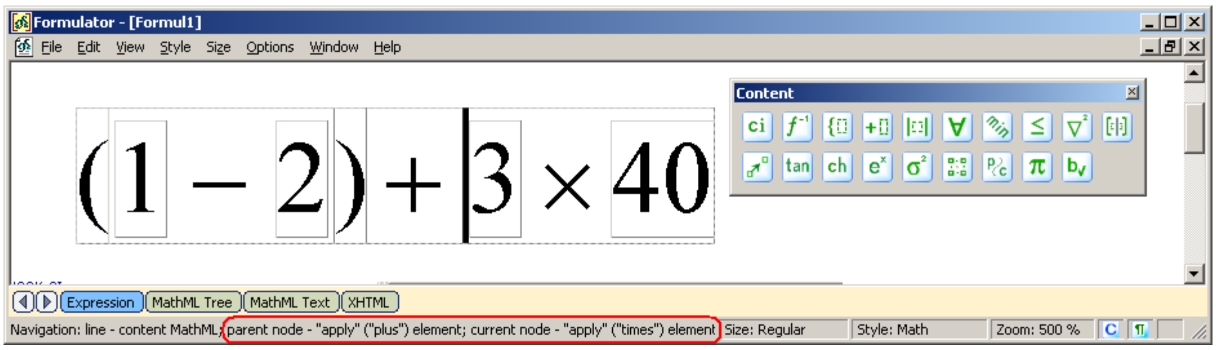

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Statistics
Expression MathML Tree MathML Text XHTML

Statistics | Size: Regular | Style: Math | Zoom: 500 %

Press the Left arrow three times. We now again on the level of the <times/> “apply” element. In this position we could delete the last argument after pressing Delete button two times. The first pressure would select the entire input slot and mark it with black. The second pressure would confirm deletion.

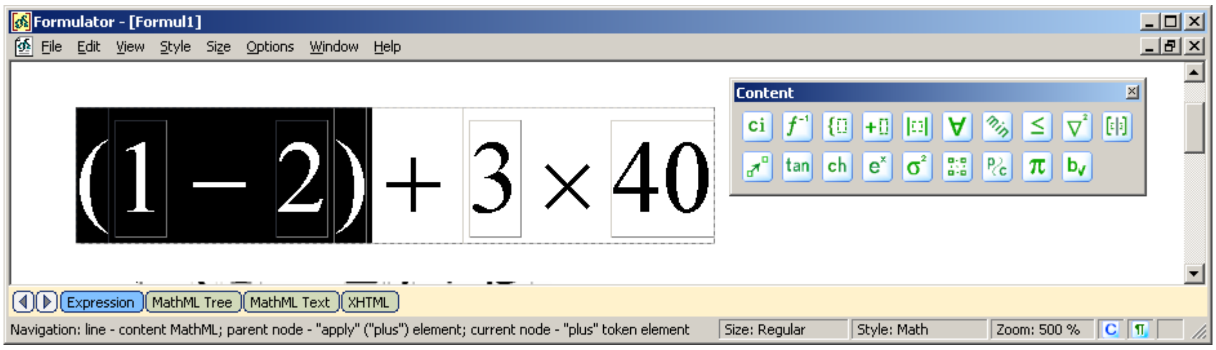
In order to get back to the higher level of the <plus/> “apply” element we need now 5 times pressing of the Left arrow. By each pressing we go through different levels of the formula structure, where each level propose its own editing facilities. Imagine that we are navigating through the tree of Content Markup elements and this procedure becomes evident.

See how the navigation information bar helps to understand which level of the formula structure is currently available for editing.



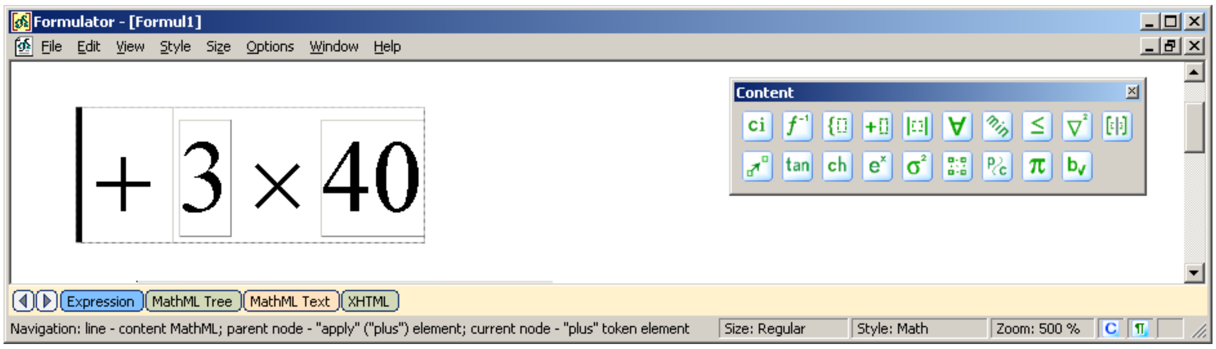
Navigation: line - content MathML; parent node - "apply" ("plus") element; current node - "apply" ("times") element | Size: Regular | Style: Math | Zoom: 500 %

Press the Left arrow and the Backspace button.



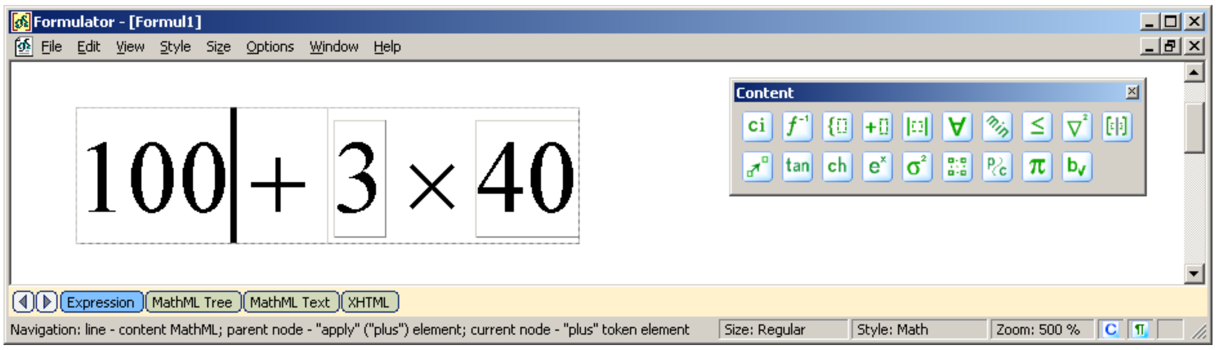
Navigation: line - content MathML; parent node - "apply" ("plus") element; current node - "plus" token element | Size: Regular | Style: Math | Zoom: 500 %

(1 – 2) form is marked for deletion. Press the Backspace button once more time.

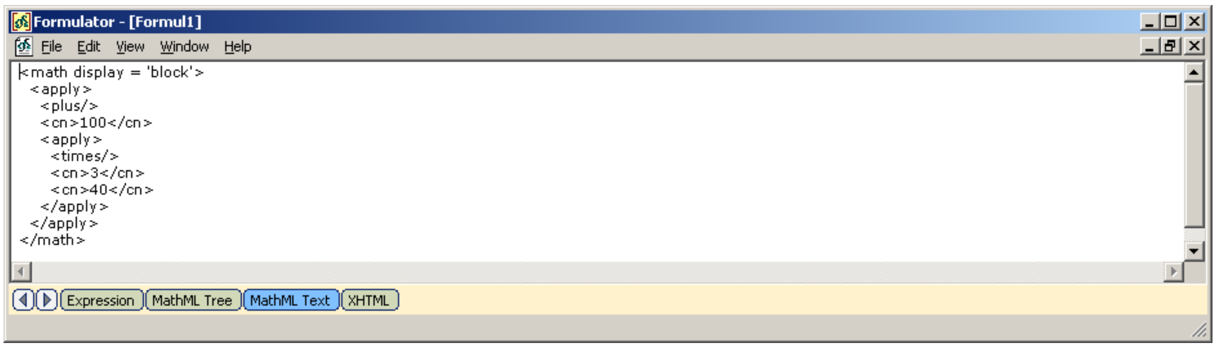


Navigation: line - content MathML; parent node - "apply" ("plus") element; current node - "plus" token element | Size: Regular | Style: Math | Zoom: 500 %

Now type number 100 and by switching to the "MathML Text" see how MathML Weaver automatically detected that while a user is situated inside of the Content Markup expression then the Content token element is needed.



Navigation: line - content MathML; parent node - "apply" ("plus") element; current node - "plus" token element | Size: Regular | Style: Math | Zoom: 500 %



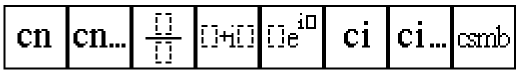
```
<math display = 'block'>
 <apply>
  <plus/>
  <cn>100</cn>
  <apply>
   <times/>
   <cn>3</cn>
   <cn>40</cn>
  </apply>
 </apply>
</math>
```

Formulator - [Formul1]
File Edit View Window Help
Expression MathML Tree MathML Text XHTML

Token Elements: externally defined symbol (csymbol)

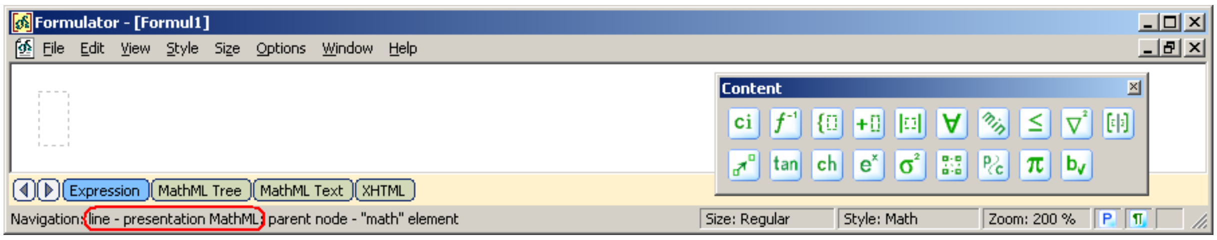
Content expression trees are built up starting from token elements. Numbers and symbols are marked by the "cn" and "ci" elements; the "csymbol" element allows to create an element whose semantics are externally defined.

Content Markup token elements are available via ci toolbar:



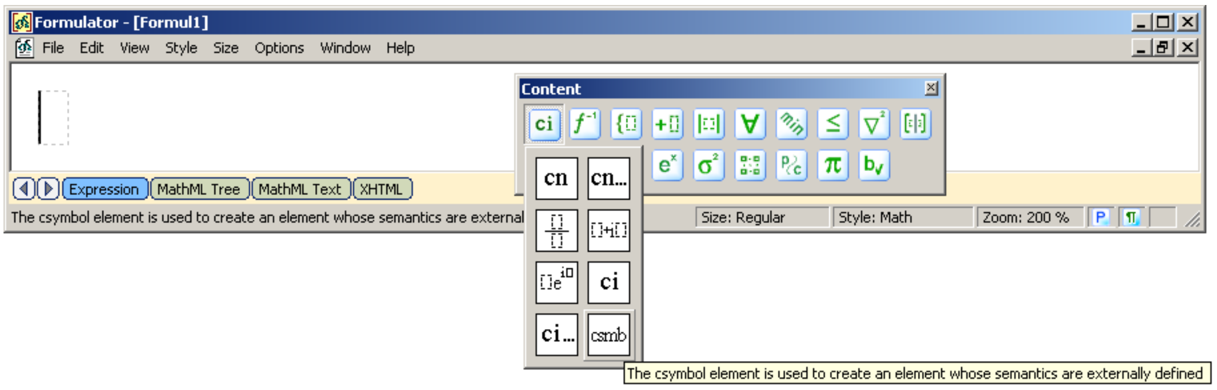
As we can see from this figure, MathML Weaver provide several buttons for a token element to account for existing options of base, type, etc. A button with '…' sign on its image leads to a dialog requesting some context dependent attributes of the newly created expression (e.g., type of an identifier).

1. The simplest case of the Content token element is presented by "csymbol". When a user presses csmb icon, the current empty slot is not altered visually, but the navigation information on the status bar helps to understand that we have already started to build a Content expression tree. The next tree figures shows this in details.



Navigation: line - presentation MathML; parent node - "math" element

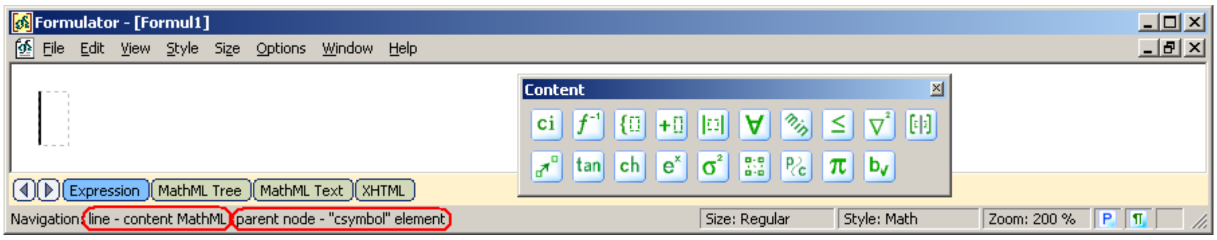
Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %



The csymbol element is used to create an element whose semantics are external

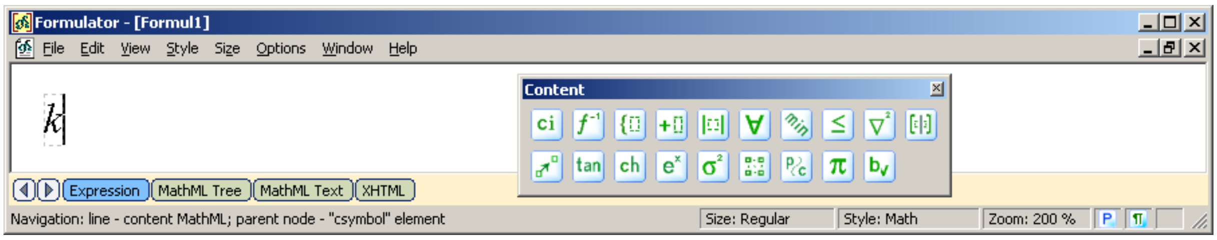
The csymbol element is used to create an element whose semantics are externally defined

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
cn
cn...
ci
ci...
csmb
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

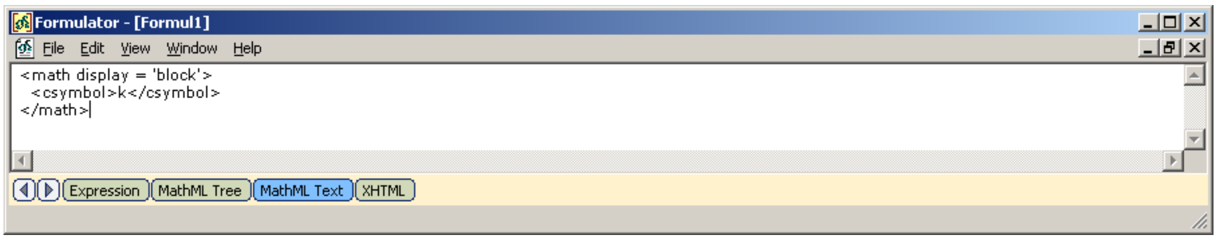


Navigation: line - content MathML; parent node - "csymbol" element

Now all that we type in the current empty slot will be interpreted as a content of the “csymbol” element:

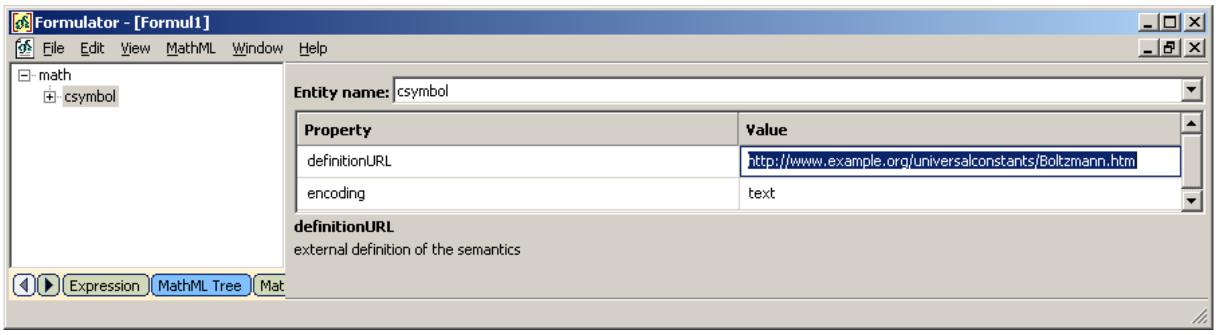


Navigation: line - content MathML; parent node - "csymbol" element



```
<math display = 'block'>
  <csymbol>k</csymbol>
</math>
```

2. It is worth noting that this just created Content expression can be supplied with all the need attributes on the page of “MathML Tree”. The next figure shows how to turn this expression into the example 4.4.1.3.3 from the W3C MathML Recommendation (reference to human readable text description of Boltzmann's constant).



Entity name: csymbol

Property	Value
definitionURL	http://www.example.org/universalconstants/Boltzmann.htm
encoding	text

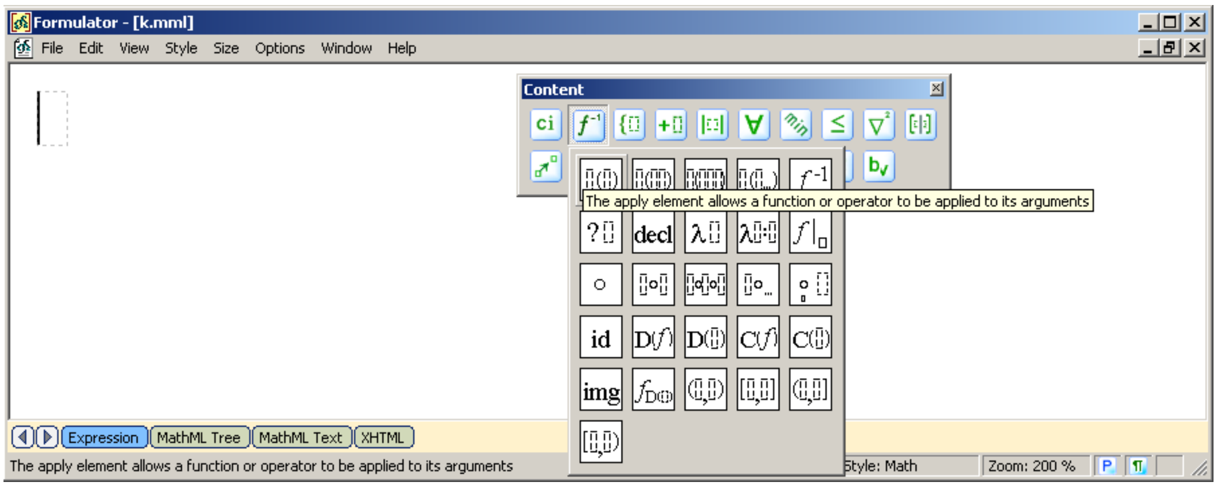
definitionURL
external definition of the semantics

3. Another important feature of the inserted visual forms for Content Markup in MathML Weaver is that they also can carry Presentation Markup. For example, if we use a set of Presentation toolbars we can easily create another example from the section 4.4.1.3.3 of the W3C MathML Recommendation (reference to OpenMath formal syntax definition of Bessel function):

```
<!-- reference to OpenMath formal syntax definition of Bessel function -->
<apply>
  <csymbol encoding="OpenMath"
    definitionURL="http://www.openmath.org/cd/hypergeo2#BesselJ">
    <msub><mi>J</mi><mn>0</mn></msub>
  </csymbol>
  <ci>y</ci>
```

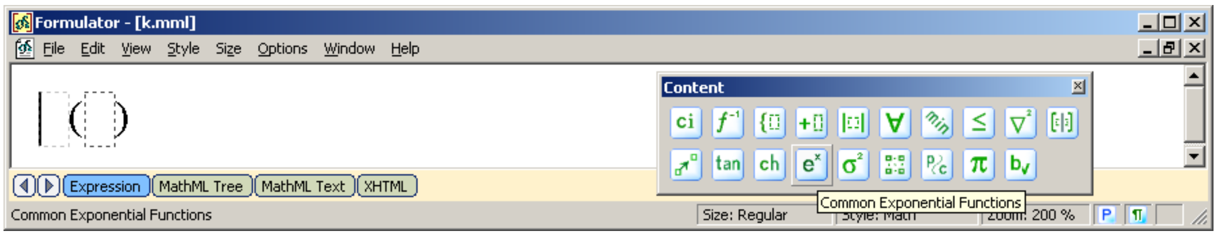
```
</apply>
```

Press the f^{-1} button on the Content toolbar set to get access to the group of "apply" elements of different form. Choose the first button.


Formulator - [k.mml]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Style: Math
Zoom: 200 %

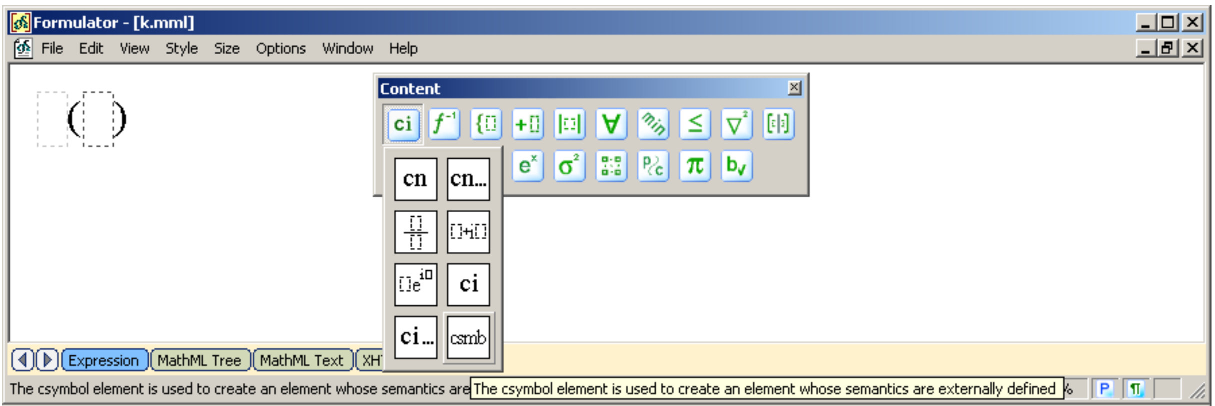
The apply element allows a function or operator to be applied to its arguments

This will insert an "apply" element with one argument in its functional form.

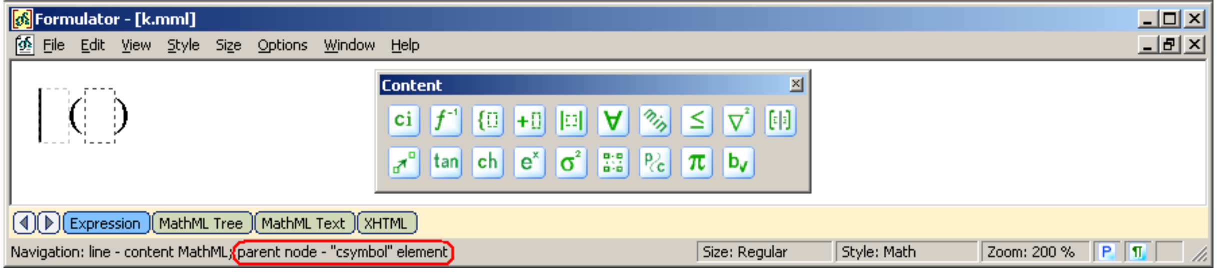

Formulator - [k.mml]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

Common Exponential Functions

Now insert the "csymbol" element in the first input slot of the "apply" element.

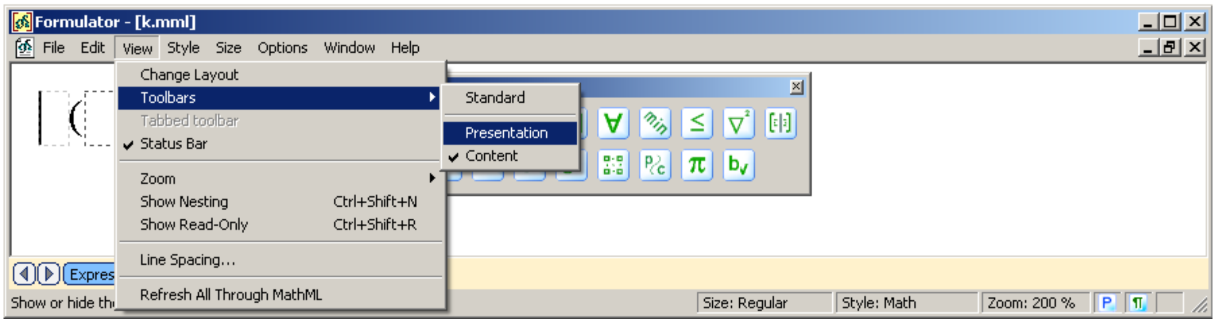

Formulator - [k.mml]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text

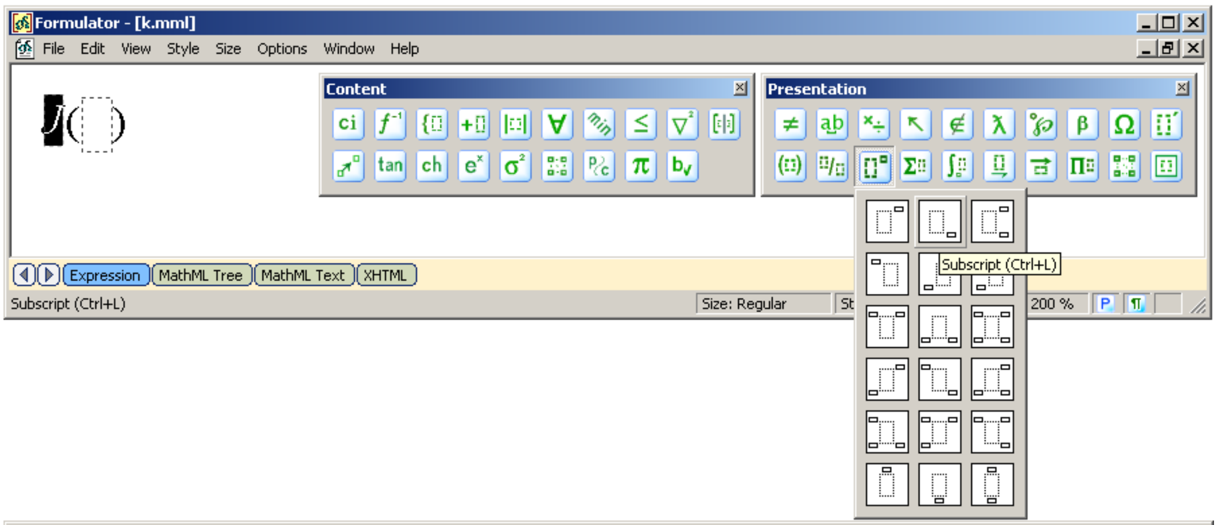
The csymbol element is used to create an element whose semantics are externally defined

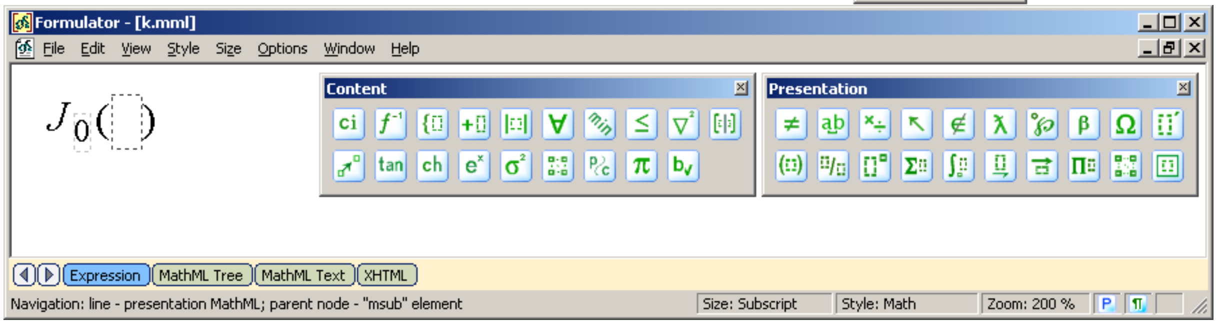

Formulator - [k.mml]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

Navigation: line - content MathML; parent node - "csymbol" element

Using the Presentation toolbars set build up contents of the "csymbol" element: <msub><mi>J</mi><mn>0</mn></msub> (J_0).

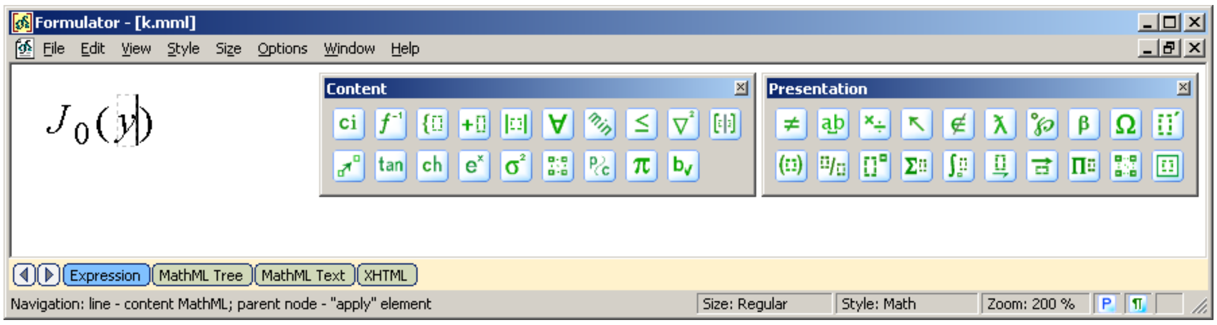






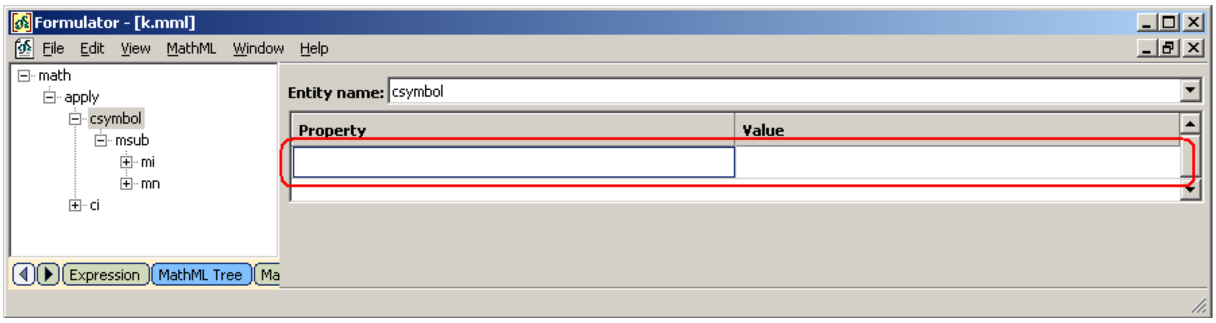
Navigation: line - presentation MathML; parent node - "msub" element Size: Subscript Style: Math Zoom: 200 %

Navigate to the second input slot of the "apply" element and press 'y'. This will fill the argument of the "apply" element with <ci>y</ci> content.

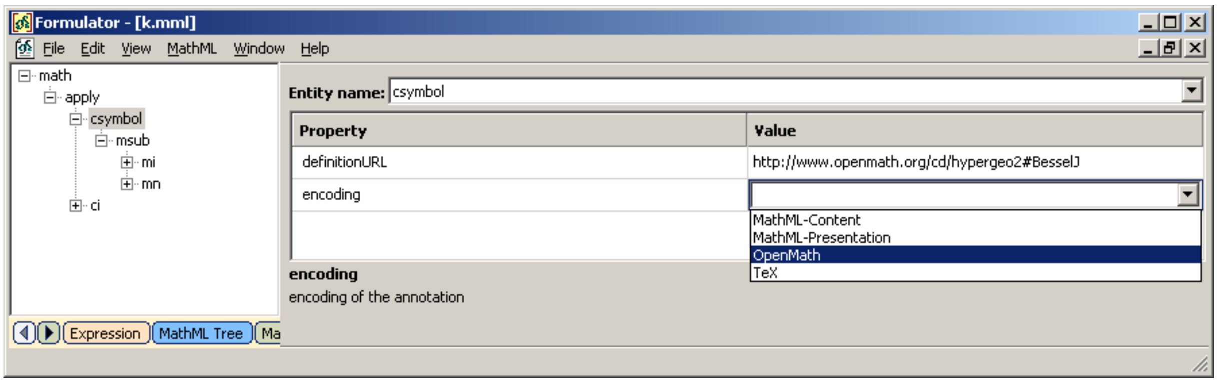


Navigation: line - content MathML; parent node - "apply" element Size: Regular Style: Math Zoom: 200 %

Now switch to the "MathML Tree" to add two attributes to the "csymbol" element. In order to do this click on the right side of the document ("Property-Value" dialog) and press the Insert button. We now see that an empty attribute record is added.



Using drop-down lists create "encoding" and "definitionURL" attributes.



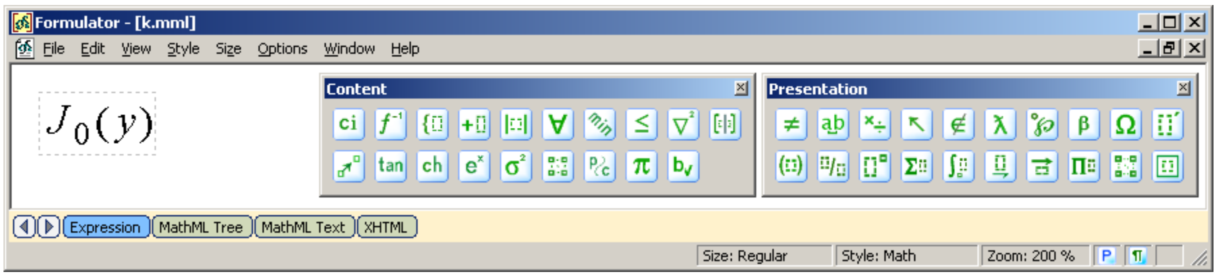
Entity name: csymbol

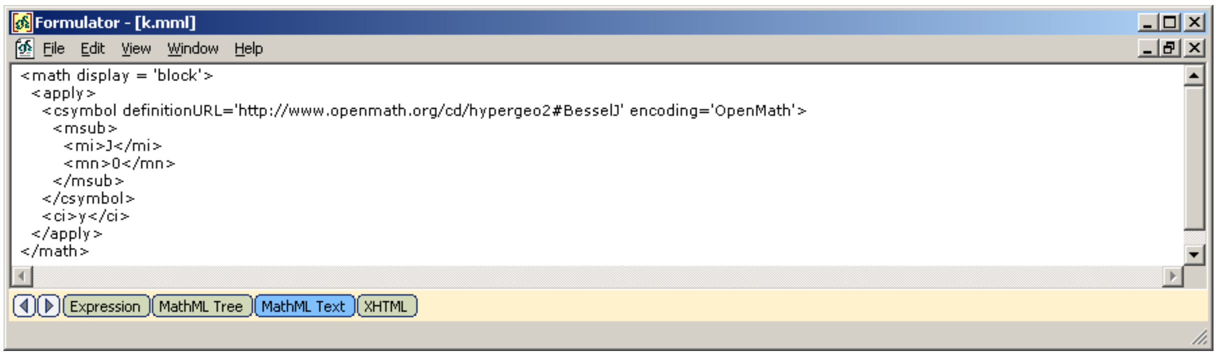
Property	Value
definitionURL	http://www.openmath.org/cd/hypergeo2#BesselJ
encoding	

MathML-Content
MathML-Presentation
OpenMath
TeX

encoding
encoding of the annotation

Check the results on the "Expression" and "MathML Text" pages.


Formulator - [k.mml]
File Edit View Style Size Options Window Help
$J_0(y)$
Content
Presentation
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

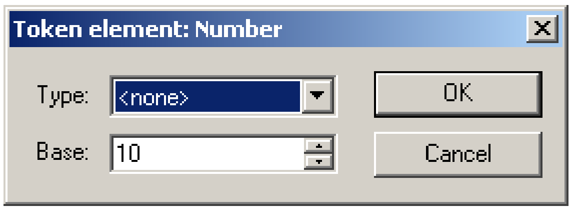

Formulator - [k.mml]
File Edit View Window Help
Expression MathML Tree MathML Text XHTML

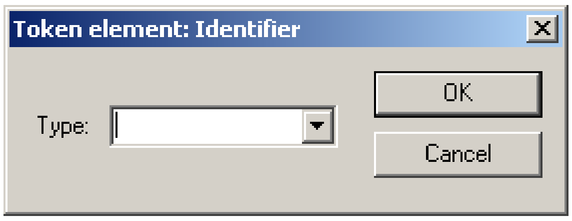
```
<math display = 'block'>
 <apply>
  <csymbol definitionURL='http://www.openmath.org/cd/hypergeo2#BesselJ' encoding='OpenMath'>
   <msub>
    <mi>J</mi>
    <mn>0</mn>
   </msub>
  </csymbol>
  <ci>y</ci>
 </apply>
</math>
```

Token Elements: numbers (cn) and identifiers (ci)

There are several attributes modifying Content Markup semantics for token elements. Thus the "base" attribute indicates numerical base of the number; the "type" attribute indicates type of the number or identifier. Moreover, different types of numbers will be rendered in a different way and even appearance of identifiers can be altered by changing the "type" attribute.

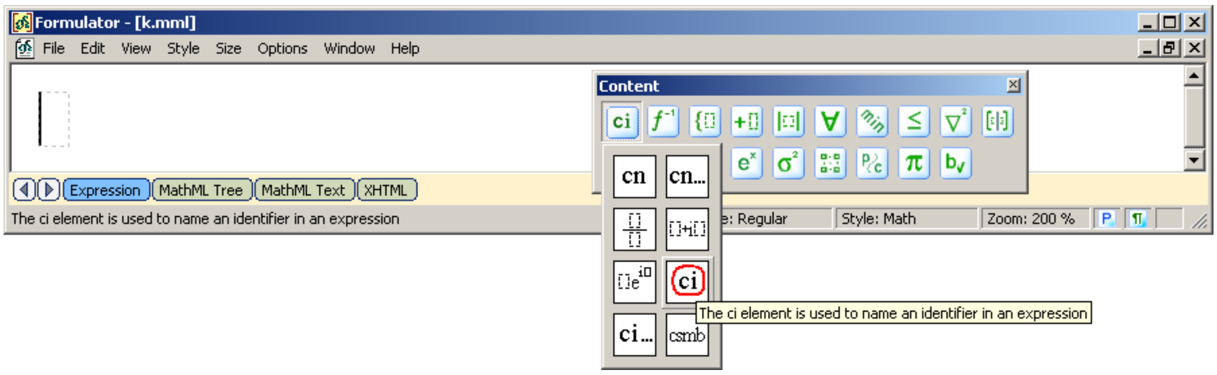
In order to visually support these requirements, Formulator 3.7 MathML Weaver proposes several ways to input Content Markup token elements: namely, a user can choose a button representing the needed type of a number from the toolbar or make some alterations in the attribute values via the "MathML Tree" page or by using corresponding property dialogs for buttons cn... and ci...:


Token element: Number
Type: <none>
Base: 10
OK
Cancel

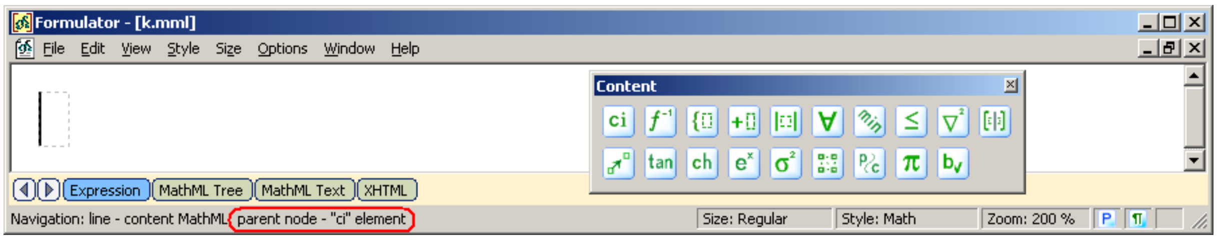

Token element: Identifier
Type:
OK
Cancel

1. The next example shows how to create a simple identifier token element and how to change its appearance by editing the "type" attribute.

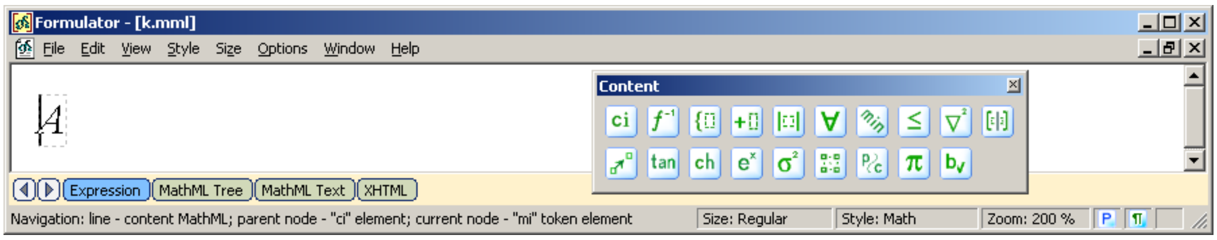
Insert the "ci" element by pressing the ci button.


Formulator - [k.mml]
File Edit View Style Size Options Window Help
Content
Expression | MathML Tree | MathML Text | XHTML
The ci element is used to name an identifier in an expression
Size: Regular | Style: Math | Zoom: 200 %
The ci element is used to name an identifier in an expression

As it is shown on the next figure, the "ci" element appears as an empty slot and the navigation bar information makes sure that it is inserted.


Formulator - [k.mml]
File Edit View Style Size Options Window Help
Content
Expression | MathML Tree | MathML Text | XHTML
Navigation: line - content MathML parent node - "ci" element
Size: Regular | Style: Math | Zoom: 200 %

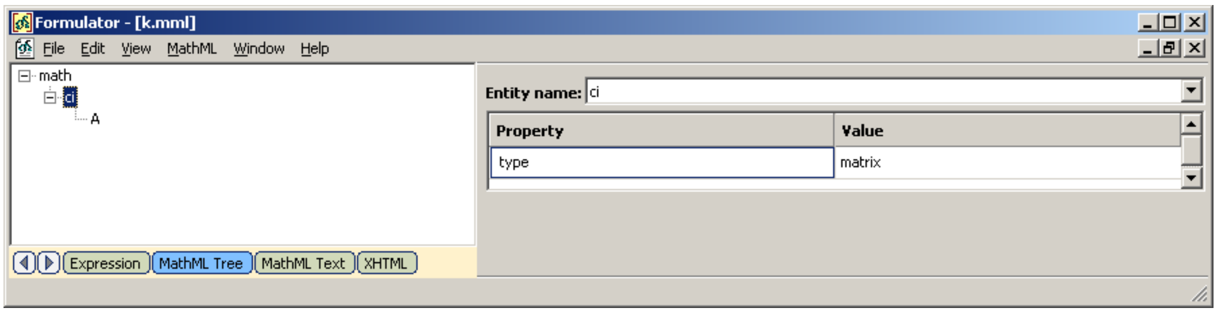
Now type an identifier in the current input slot. Note that the context of the "ci" element is one of three cases where presentation markup may appear in content markup properly. So, not only plain text can be typed into the "ci" input slot, but also some kind of presentation markup tree.


Formulator - [k.mml]
File Edit View Style Size Options Window Help
Content
ci
tan
ch
Expression MathML Tree MathML Text XHTML

Navigation: line - content MathML; parent node - "ci" element; current node - "mi" token element | Size: Regular | Style: Math | Zoom: 200 %

According to the default value of the "type" attribute of the "ci" element (unspecified type), contents of the "ci" element are represented as if it were contents of a "mi" element (italic style).

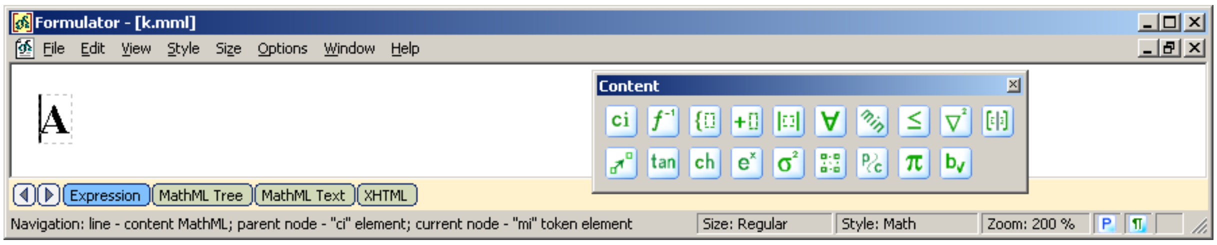
Now change the current page of MathML Weaver and add the "type" attribute into the "ci" node. Then choose the "matrix" value from the list of known identifier's types.


Formulator - [k.mml]
File Edit View MathML Window Help
math
ci
A
Expression MathML Tree MathML Text XHTML

Entity name: ci

Property	Value
type	matrix

Switch back to the "Expression" page and see how rendering of the "ci" element is changed. Now *A* is bold according to the typographical rules of representing the "matrix" type.

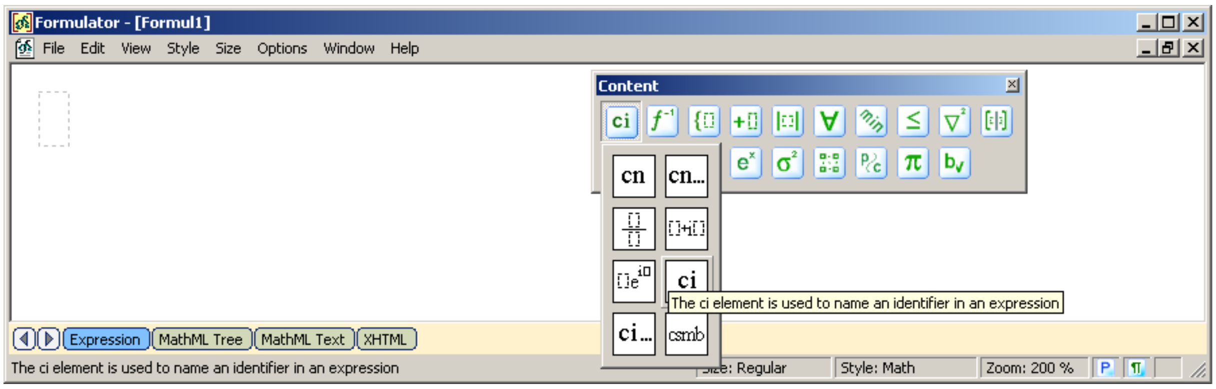

Formulator - [k.mml]
File Edit View Style Size Options Window Help
Content
ci
tan
ch
Expression MathML Tree MathML Text XHTML

Navigation: line - content MathML; parent node - "ci" element; current node - "mi" token element | Size: Regular | Style: Math | Zoom: 200 %

As in the case of the "csymbol" element, Content Markup identifiers can carry Presentation Markup. For example, if we use a set of Presentation toolbars we can create an example from the section 4.4.1.2.2 of the W3C MathML Recommendation:

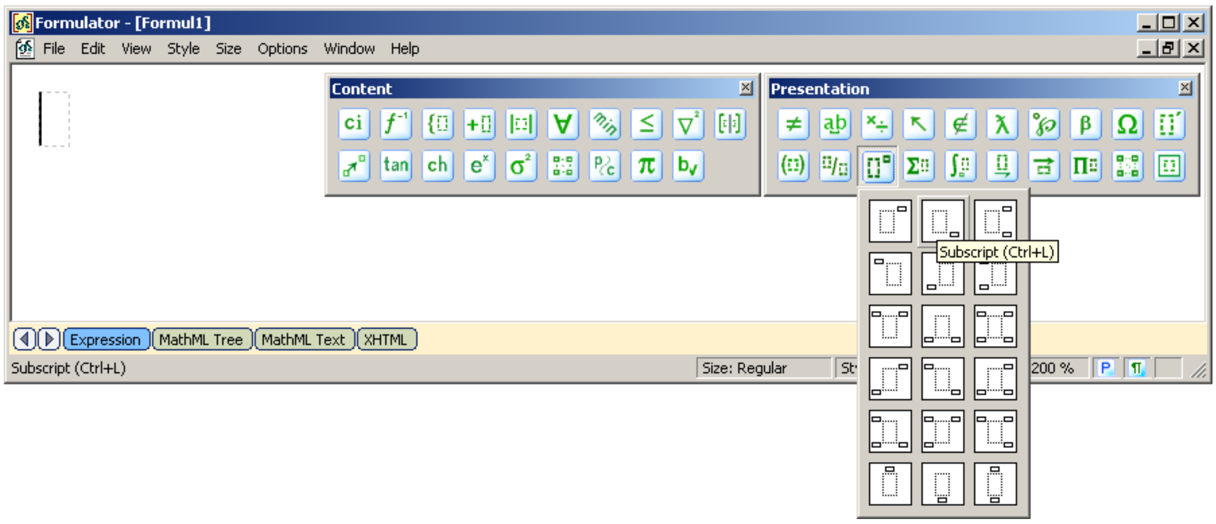
```
<ci>
  <msub>
    <mi>x</mi>
    <mi>i</mi>
  </msub>
</ci>
```

Insert the "ci" element.

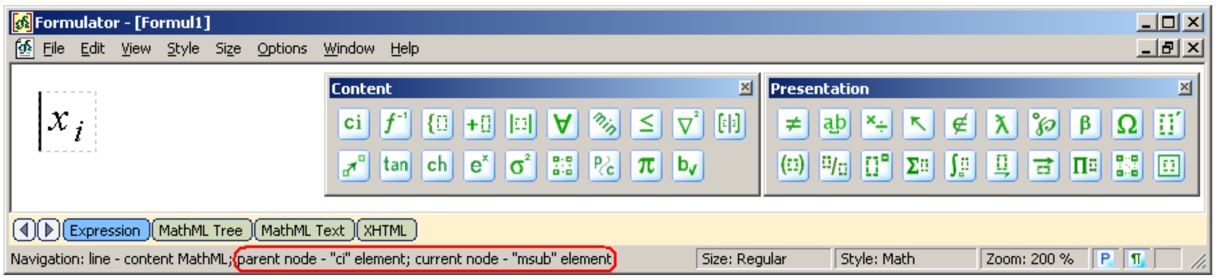

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
cn
cn...
ci
ci...
csmb
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

The ci element is used to name an identifier in an expression

Use the Presentation toolbar to insert Subscript node and fill it with x_i expression.

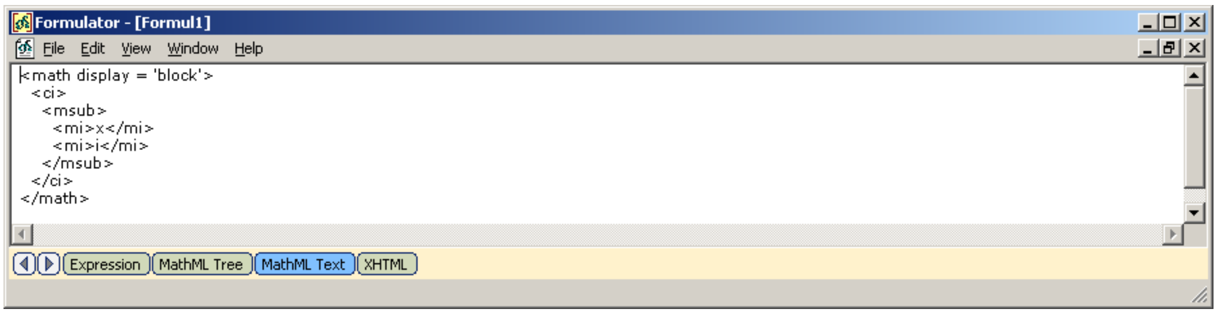

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Presentation
Subscript (Ctrl+L)
Expression MathML Tree MathML Text XHTML
Size: Regular
200 %

Subscript (Ctrl+L)


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Presentation
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

Navigation: line - content MathML; parent node - "ci" element; current node - "msub" element

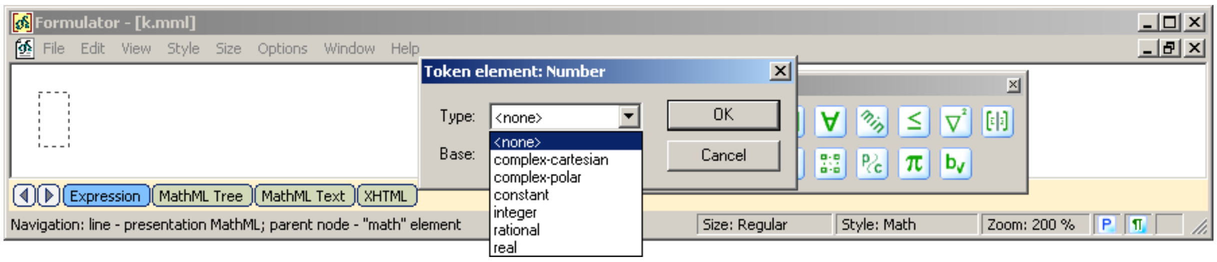
See results on the “MathML Text” page.


Formulator - [Formul1]
File Edit View Window Help
Expression MathML Tree MathML Text XHTML

```
<math display = 'block'>
  <ci>
    <msub>
      <mi>x</mi>
      <mi>i</mi>
    </msub>
  </ci>
</math>
```

2. The next example shows how to create a simple number token element and how to control its appearance and semantics by altering "type" and "base" attributes.

Insert the "cn" element by pressing the cn... button. Then the initial property dialog for the "cn" element will be shown.


Formulator - [k.mml]

Token element: Number

Type: <none>

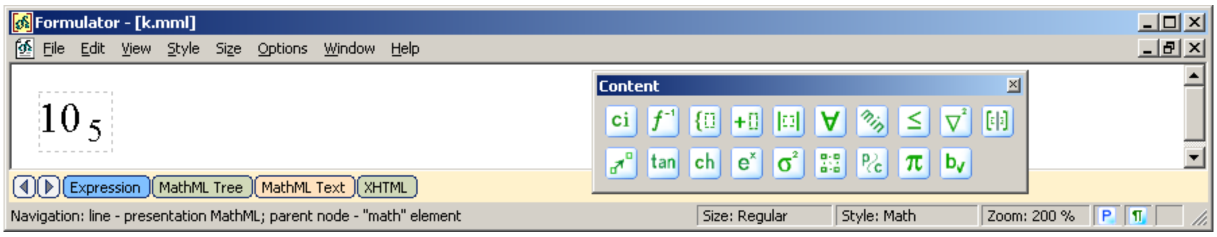
Base:

<none>
complex-cartesian
complex-polar
constant
integer
rational
real

Navigation: line - presentation MathML; parent node - "math" element

Size: Regular Style: Math Zoom: 200 %
File Edit View Style Size Options Window Help
OK
Cancel
Expression MathML Tree MathML Text XHTML

Choose the "integer" value in the "type" drop-down list and 5 as a base of a number. This feature lets creating numbers of different numerical bases.


Formulator - [k.mml]

10_5

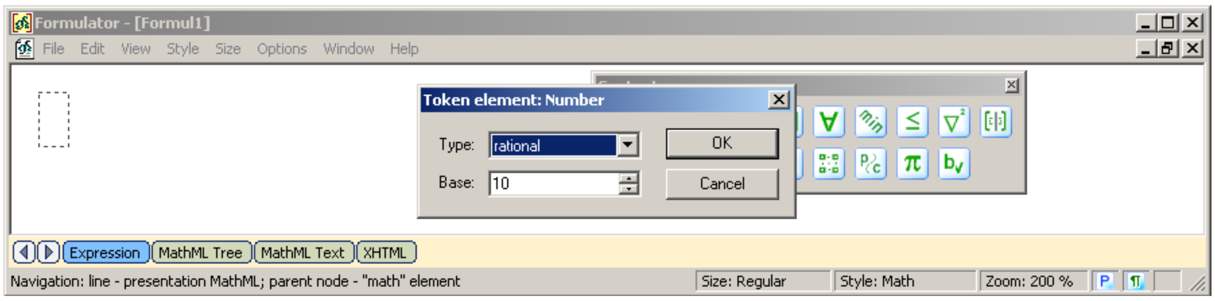
Navigation: line - presentation MathML; parent node - "math" element

Size: Regular Style: Math Zoom: 200 %
File Edit View Style Size Options Window Help
Content
ci
tan
ch
Expression MathML Tree MathML Text XHTML

A user can control number's type not only via different values proposed by the initial property dialog ("rational", "complex-cartesian", "complex-polar"), but also with the help other available buttons of the ci toolbar.

For example, rational numbers can be created by pressing the cn... button if the "rational" type is selected afterwards from the initial property dialog or at once with the button.

Press the cn... button and select the "rational" type.


Formulator - [Formul1]

Token element: Number

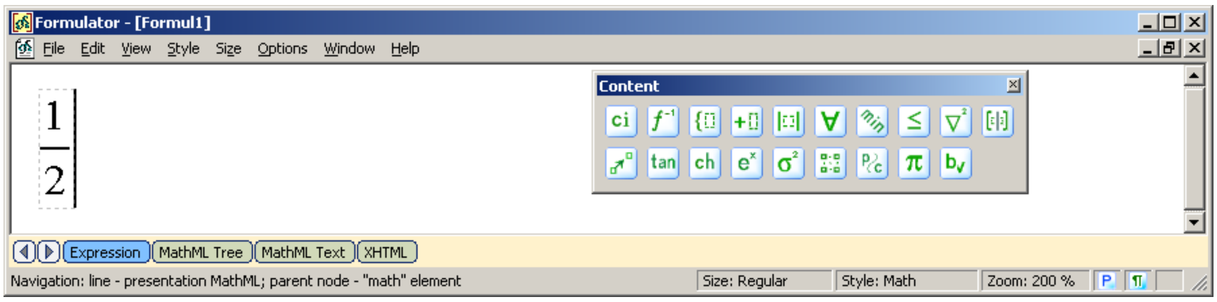
Type: rational

Base: 10

Navigation: line - presentation MathML; parent node - "math" element

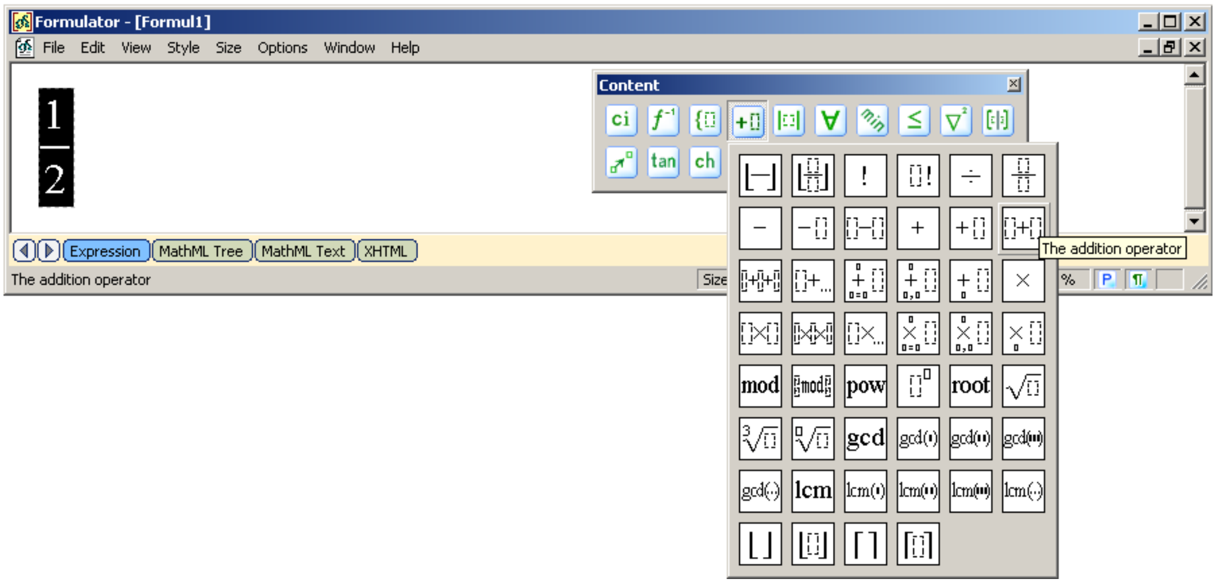
Size: Regular Style: Math Zoom: 200 %
File Edit View Style Size Options Window Help
OK
Cancel
Expression MathML Tree MathML Text XHTML

Fill numerator and denominator with the numerical values.


Formulator - [Formul1]

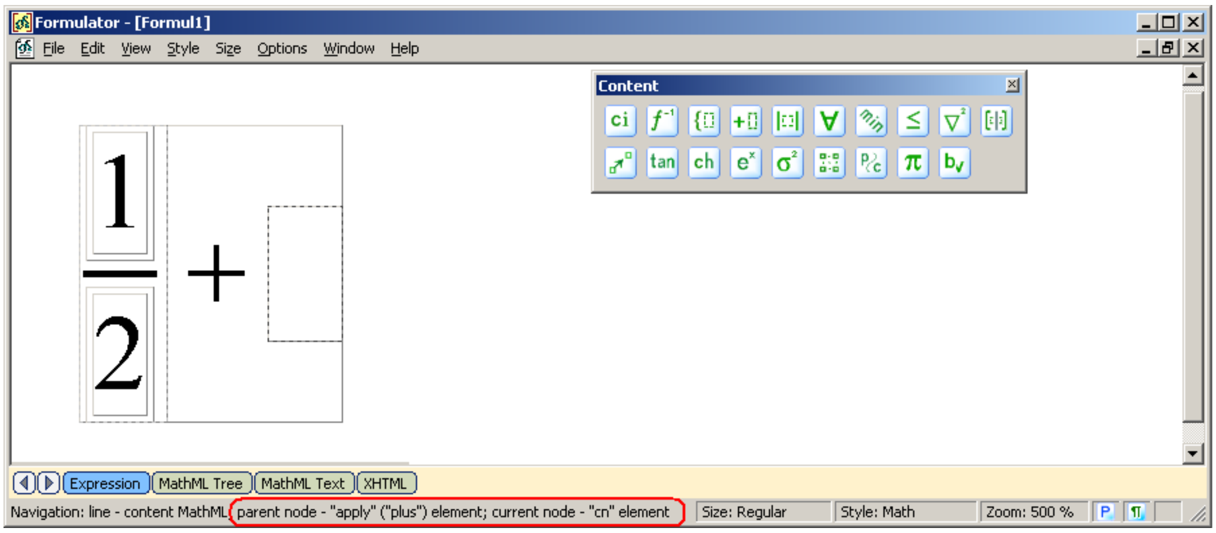
Navigation: line - presentation MathML; parent node - "math" element | Size: Regular | Style: Math | Zoom: 200 %

Select the created rational number (with Shift + arrow keys or with mouse) and click on the +[] button. The pane of Arithmetic Operators of the Content Markup is now available. After pressing on one of the buttons representing the "apply" element with one and more arguments the current selection will be considered as the first argument of this "apply" element.


Formulator - [Formul1]

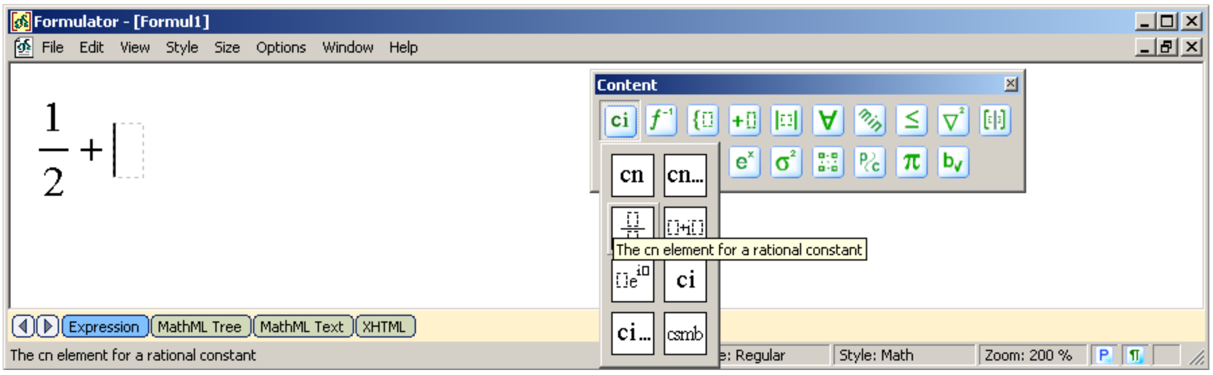
The addition operator

Let's press on the []+[] button and see how the structure of the built MathML tree is changed. The figure suggests that the <apply><plus/>…</apply> pattern is inserted and the rational "cn" element is the first argument of this operation.


Formulator - [Formul1]

Navigation: line - content MathML parent node - "apply" ("plus") element; current node - "cn" element | Size: Regular | Style: Math | Zoom: 500 %

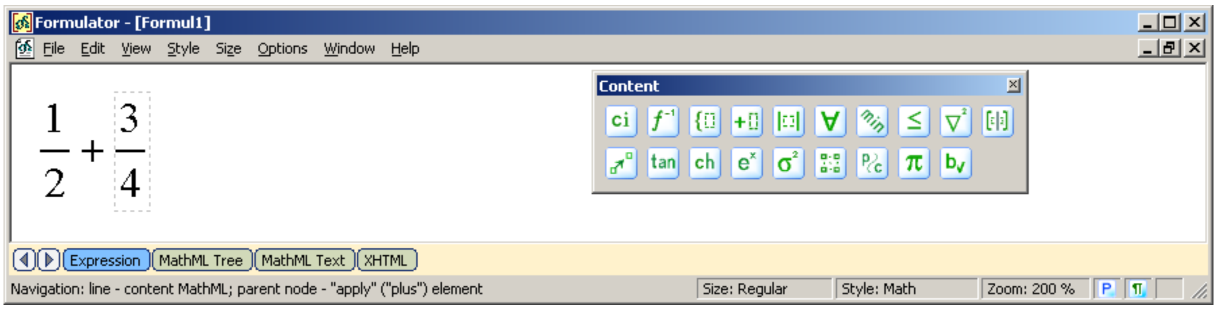
To fill the second input slot, navigate to it with the Right arrow and press the button of the ci toolbar. This will insert another rational number without additional dialogs.



$\frac{1}{2} + \square$

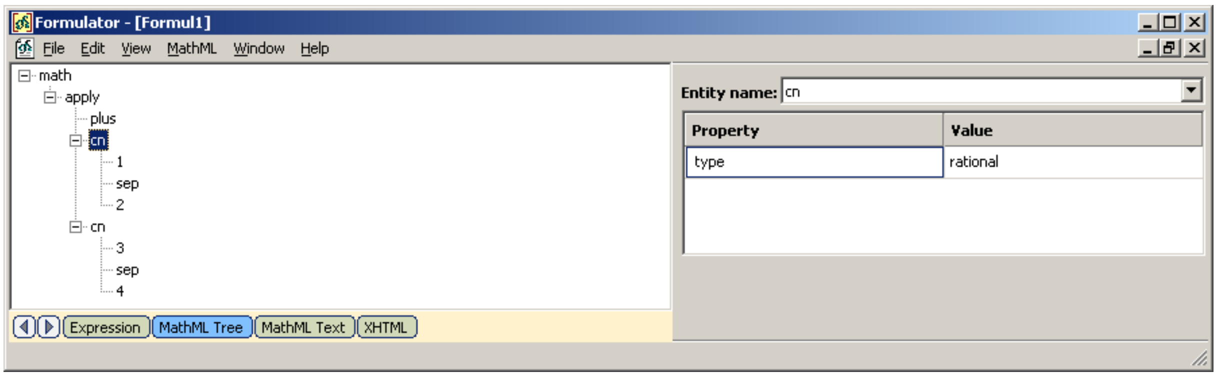
The cn element for a rational constant

Compare results on different pages of MathML Weaver (“Expression”, “MathML Tree”, “MathML Test”).



$\frac{1}{2} + \frac{3}{4}$

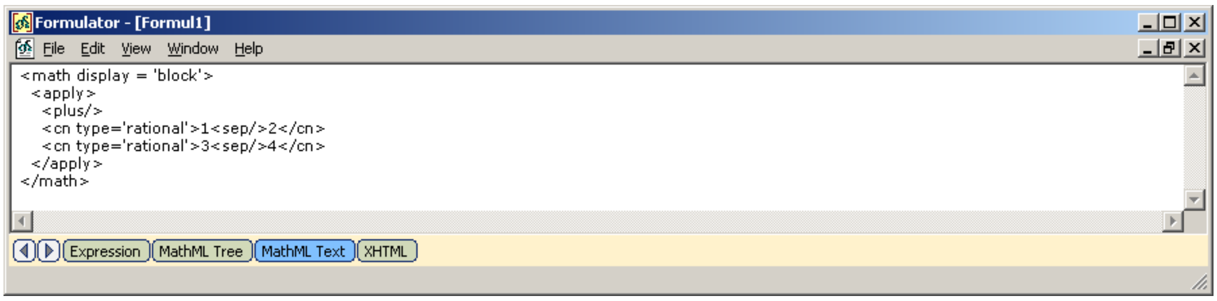
Navigation: line - content MathML; parent node - "apply" ("plus") element



```
math
  apply
    plus
    cn
      1
      sep
      2
    cn
      3
      sep
      4
```

Entity name: cn

Property	Value
type	rational

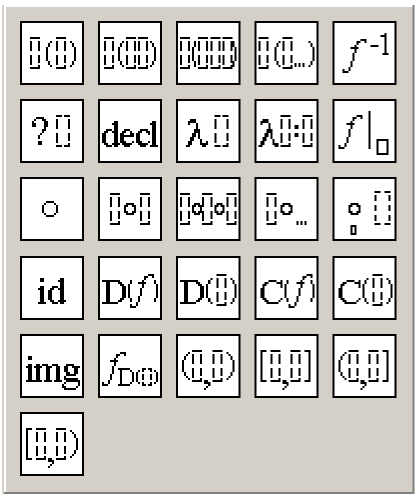


```
<math display = 'block'>
  <apply>
    <plus/>
    <cn type='rational'>1<sep/>2</cn>
    <cn type='rational'>3<sep/>4</cn>
  </apply>
</math>
```

Basic Content Elements: apply

The “apply” element of the Content Markup allows a function or operator to be applied to its arguments. It is the basic element in a sense that the overwhelming majority of expressions in MathML content markup is carried out by applying operators or functions to arguments.

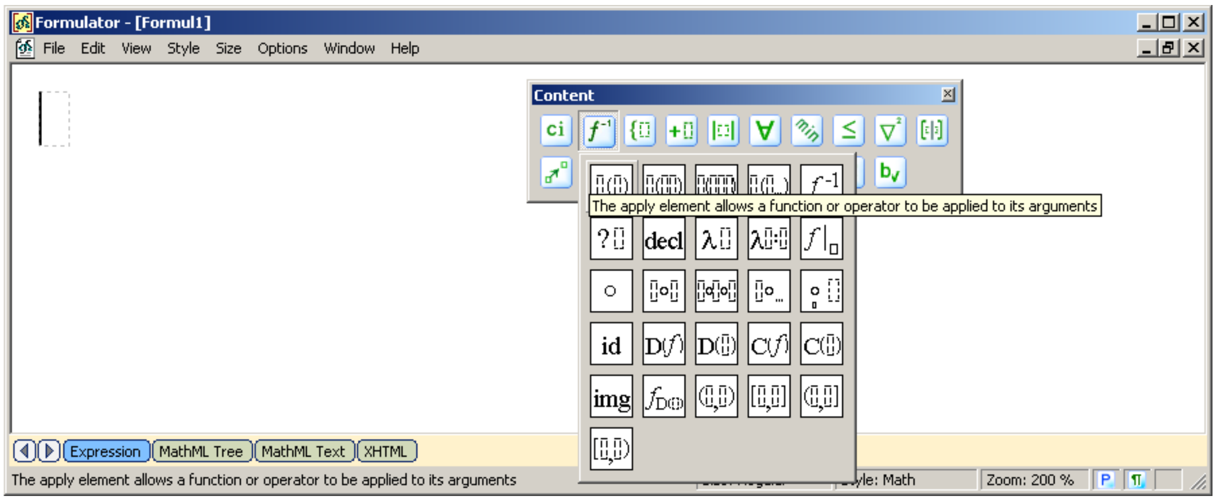
The toolbar of basic content elements f^{-1} in MathML Weaver contains several groups of buttons for creating different forms of the "apply" element and some related to this element concepts:



The first 4 buttons in the toolbar are responsible for the "apply" element itself: 1, 2, 3, and many-arguments forms. Please note, that these buttons represents the most general, functional form of the "apply" element appearance. Theoretically we can use these buttons for some other purpose, e.g., for arithmetic operators applying, but since there is a traditional infix form of rendering for arithmetic operators, such way of editing MathML content markup is not recommended. The next example shows this and others features of work with the "apply" element.

1. Insert sin(x) expression.

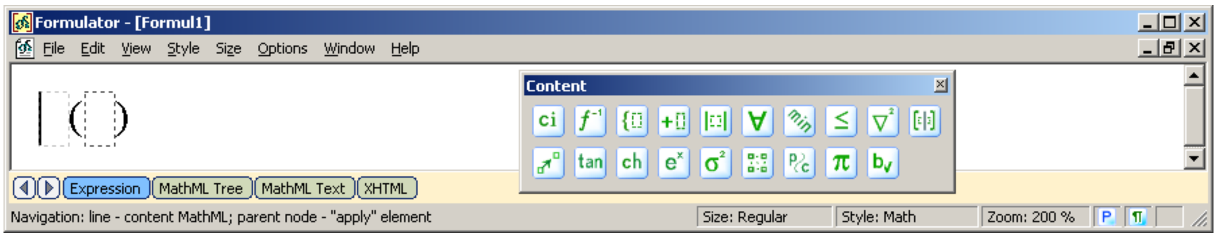
Press the ▯(▯) button; type "sin" in the first and "x" in the second input slot.

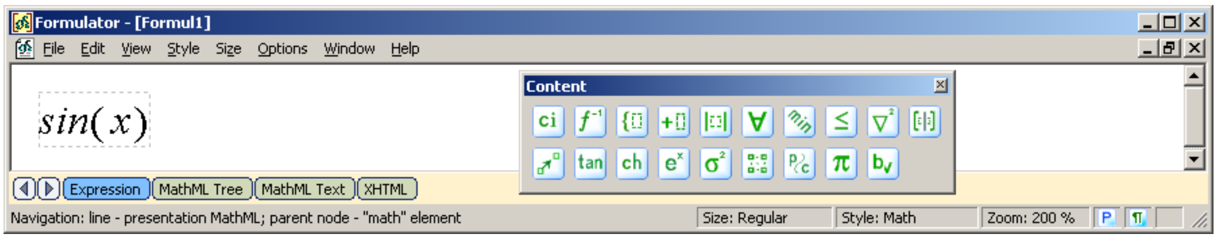


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Zoom: 200 %

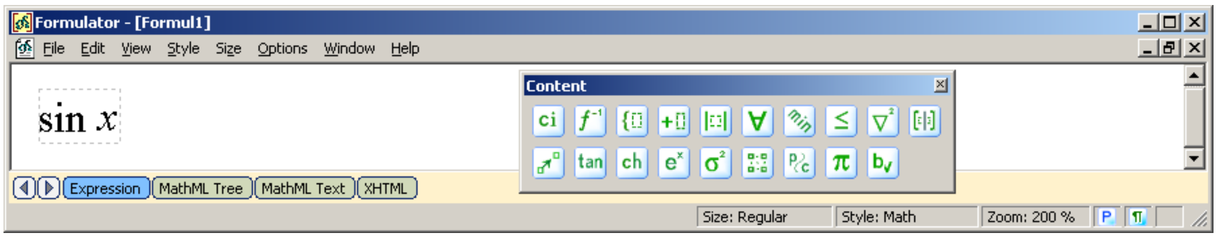
The apply element allows a function or operator to be applied to its arguments

The apply element allows a function or operator to be applied to its arguments


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular Style: Math Zoom: 200 %
Navigation: line - content MathML; parent node - "apply" element

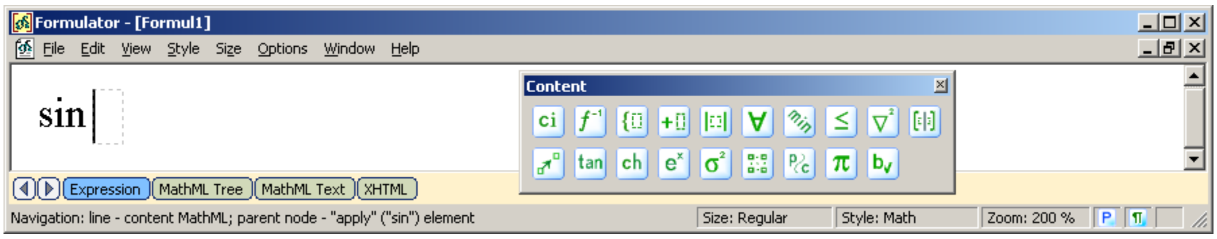

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
$sin(x)$
Expression MathML Tree MathML Text XHTML
Size: Regular Style: Math Zoom: 200 %
Navigation: line - presentation MathML; parent node - "math" element

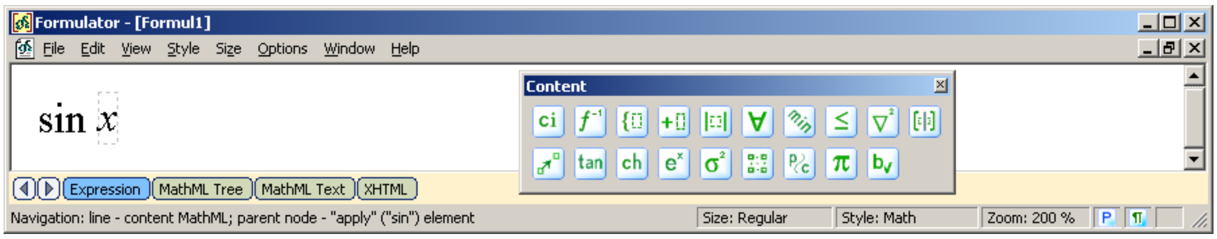
The rendering of this MathML fragment is not finished; to make it correct use the "Refresh All Through MathML" command from the "View" menu. This helps MathML Weaver to find the meaning of the entered text and to render it according to the mathematics representation rules.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
$\sin x$
Expression MathML Tree MathML Text XHTML
Size: Regular Style: Math Zoom: 200 %

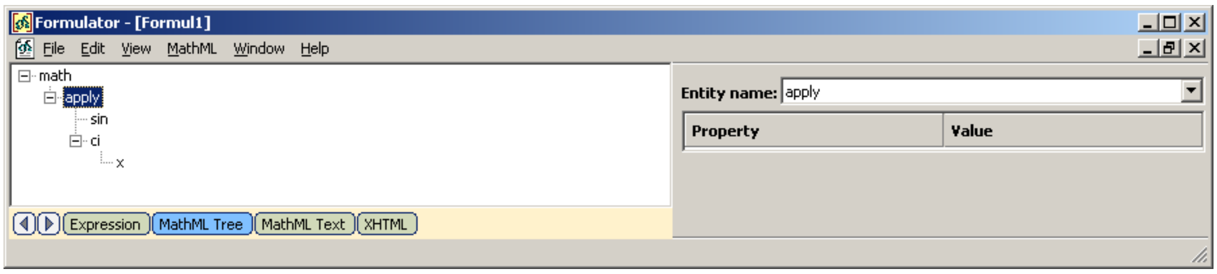
We did need such an additional action, because the most general form of the "apply" element was used. Normally, it is quite enough for users to have just special toolbars with common functions and operators. The previous example in that case would be more simple, as the next figures suggest.

Press the sin button of the common trigonometric functions toolbar (tan) and type "x" in the input slot.

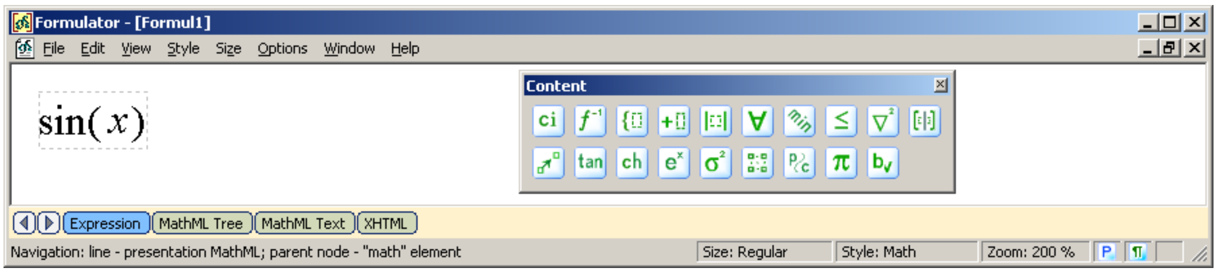

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
sin
Expression MathML Tree MathML Text XHTML
Size: Regular Style: Math Zoom: 200 %
Navigation: line - content MathML; parent node - "apply" ("sin") element


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
$\sin x$
Expression MathML Tree MathML Text XHTML
Size: Regular Style: Math Zoom: 200 %
Navigation: line - content MathML; parent node - "apply" ("sin") element

We have the same MathML tree for sin(x) as earlier, but now it was much faster.

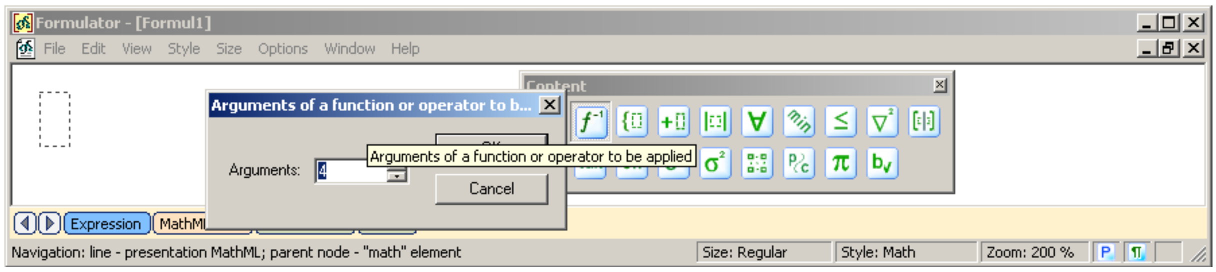

Formulator - [Formul1]
File Edit View MathML Window Help
math
apply
sin
ci
x
Entity name: apply
Property
Value
Expression MathML Tree MathML Text XHTML

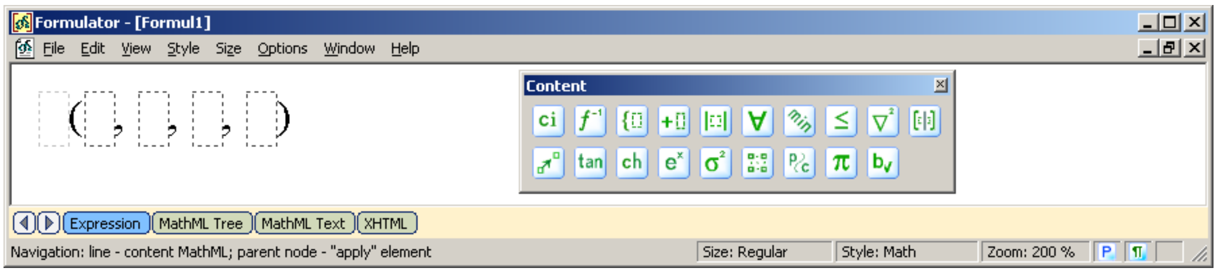
There is one more way to deal with this example, but it is similar to the first case and is shown here only for the sake of completeness. The only distinction is that we can not just type the "sin" value into the input slot, but press the corresponding sin button from the common trigonometric functions toolbar (tan). Then MathML Weaver will know about the special (plain, not italic) rendering of the first argument of the "apply" element, but still we should use the "Refresh All Through MathML" command from the "View" menu to help MathML Weaver understand that enclosing the "x" text in brackets is not needed:


Navigation: line - presentation MathML; parent node - "math" element
Formulator - [Formul1]
File Edit View Style Size Options Window Help
sin(x)
Content
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

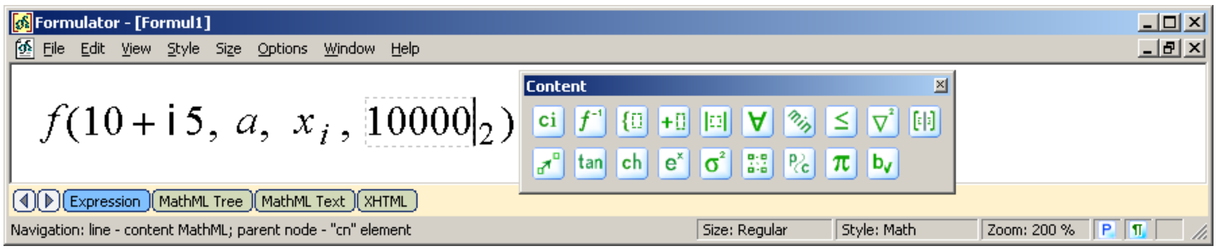
2. Insert a function of four arguments.

Press the button; fill the needed number of arguments in the shown dialog and see the newly created visual representation of the "apply" element.


Arguments of a function or operator to b...
Arguments: 4
Arguments of a function or operator to be applied
Navigation: line - presentation MathML; parent node - "math" element


Navigation: line - content MathML; parent node - "apply" element
Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

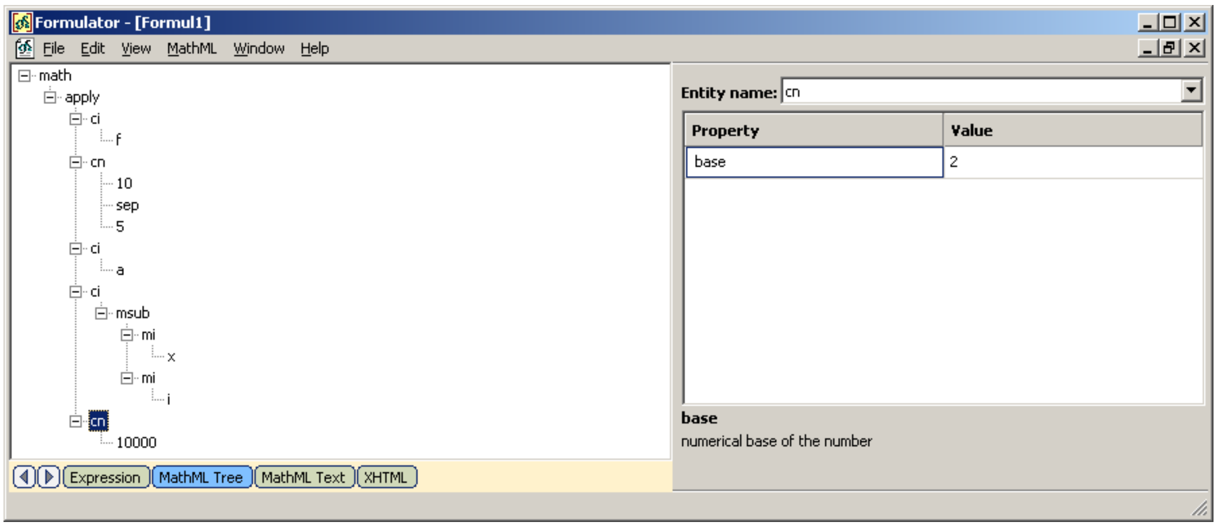
Now this form can be filled with other elements of the MathML content markup.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

$f(10+\mathrm{i}\,5,\ a,\ x_i,\ 10000_2)$

Navigation: line - content MathML; parent node - "cn" element

Size: Regular Style: Math Zoom: 200 %

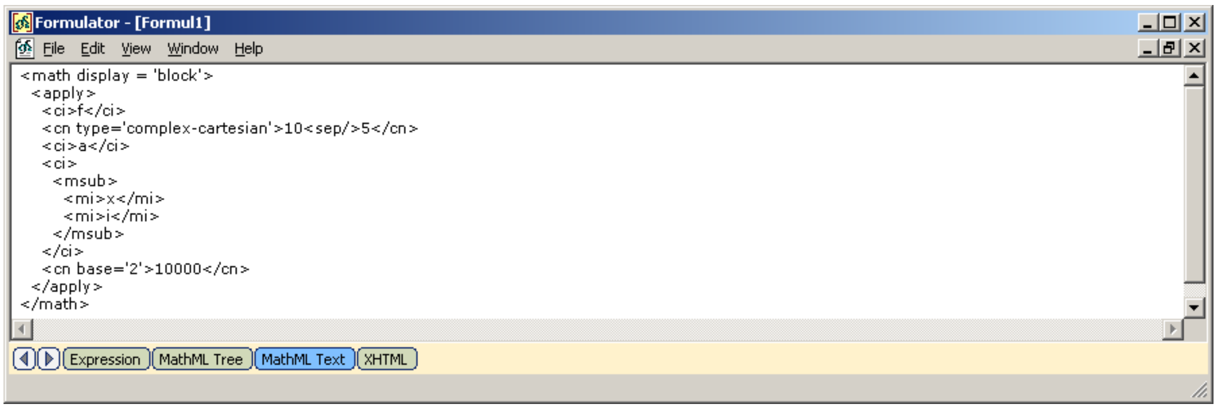

Formulator - [Formul1]
File Edit View MathML Window Help
Expression MathML Tree MathML Text XHTML

```
math
  apply
    ci
      f
    cn
      10
      sep
      5
    ci
      a
    ci
      msub
        mi
          x
        mi
          i
    cn
      10000
```

Entity name: cn

Property	Value
base	2

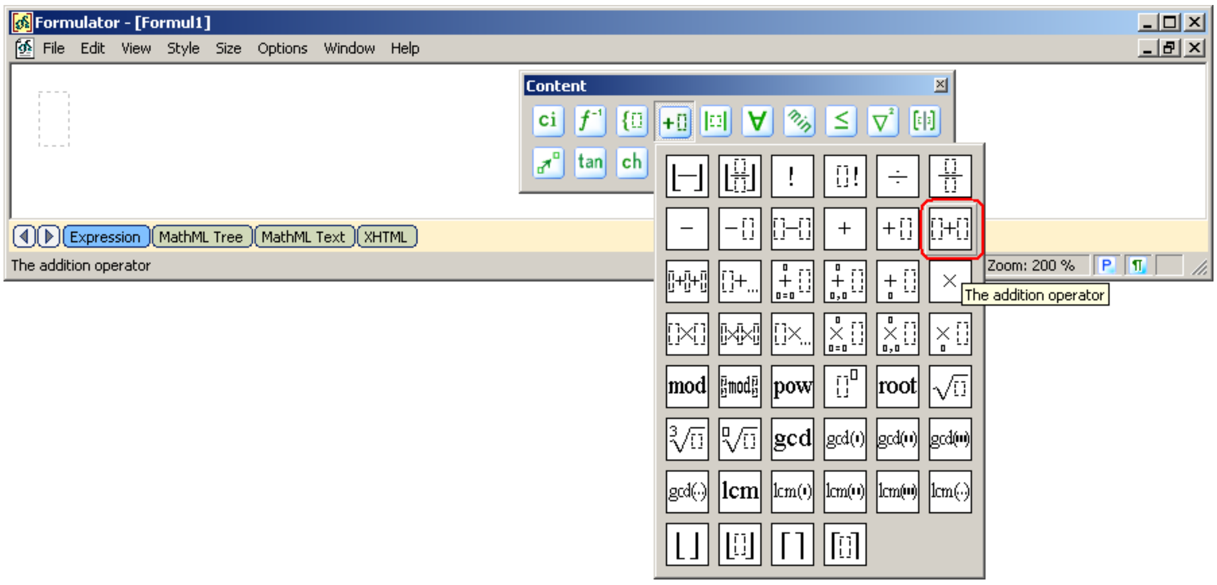
base
numerical base of the number

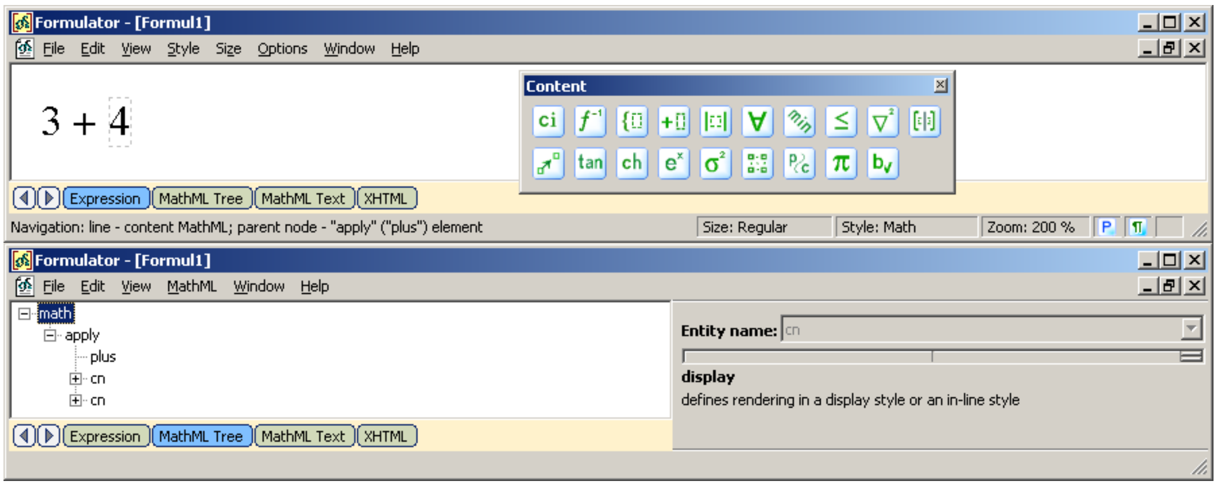

Formulator - [Formul1]
File Edit View Window Help
Expression MathML Tree MathML Text XHTML

```
<math display = 'block'>
 <apply>
  <ci>f</ci>
  <cn type='complex-cartesian'>10<sep/>5</cn>
  <ci>a</ci>
  <ci>
   <msub>
    <mi>x</mi>
    <mi>i</mi>
   </msub>
  </ci>
  <cn base='2'>10000</cn>
 </apply>
</math>
```

3. Create an arithmetic expression.

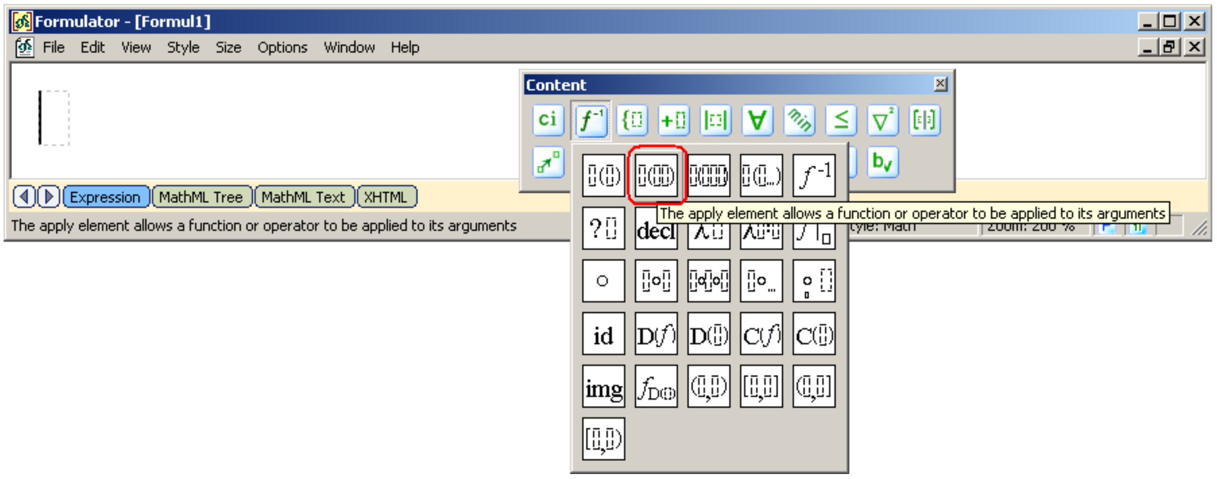
The easy way is to use the corresponding toolbar and special buttons:

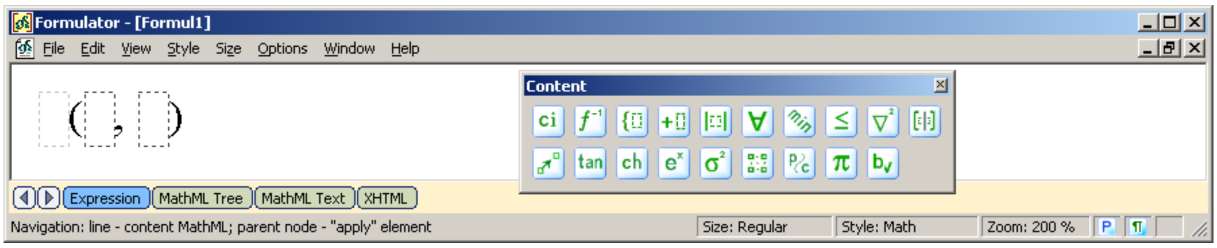

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
The addition operator
Zoom: 200 %
The addition operator


Formulator - [Formul1]
File Edit View Style Size Options Window Help
3 + 4
Content
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %
Formulator - [Formul1]
File Edit View MathML Window Help
math
apply
plus
cn
cn
Entity name: cn
display
defines rendering in a display style or an in-line style

Navigation: line - content MathML; parent node - "apply" ("plus") element

There is also a more long way to create this form by means of the “Basic Content Elements” toolbar.

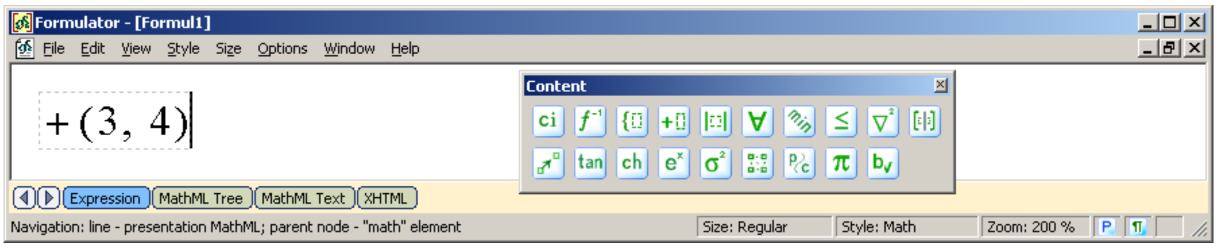

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
The apply element allows a function or operator to be applied to its arguments
The apply element allows a function or operator to be applied to its arguments


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

Navigation: line - content MathML; parent node - "apply" element

Size: Regular Style: Math Zoom: 200 %

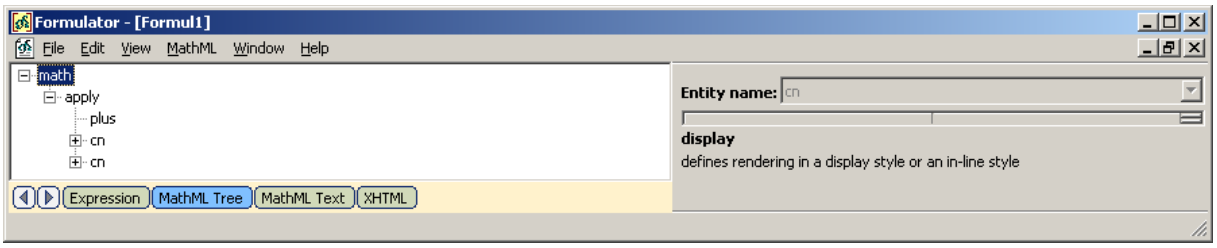
Type the operator '+' and values into input slots as if they represent the functional (in contrast to infix) notation.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
+ (3, 4)
Content
Expression MathML Tree MathML Text XHTML

Navigation: line - presentation MathML; parent node - "math" element

Size: Regular Style: Math Zoom: 200 %

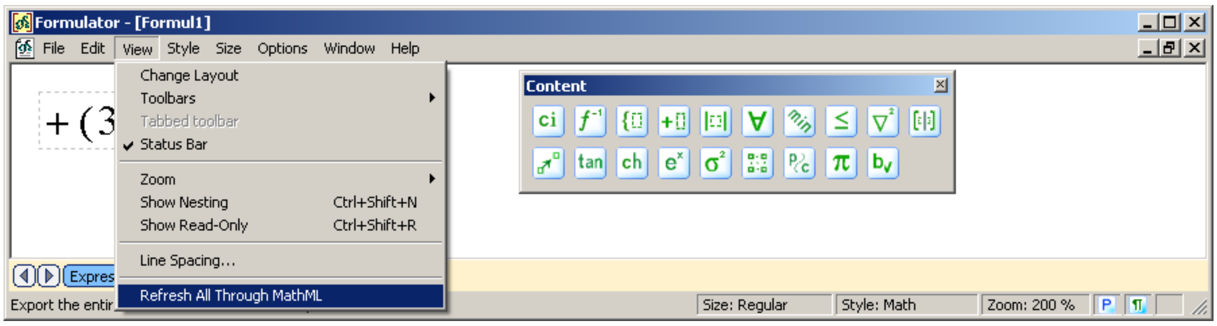
The structure of the MathML tree is already built up, but the rendering will become correct after refreshing the text by means of the "Refresh All Through MathML" command.


Formulator - [Formul1]
File Edit View MathML Window Help
math
apply
plus
cn
cn
Expression MathML Tree MathML Text XHTML

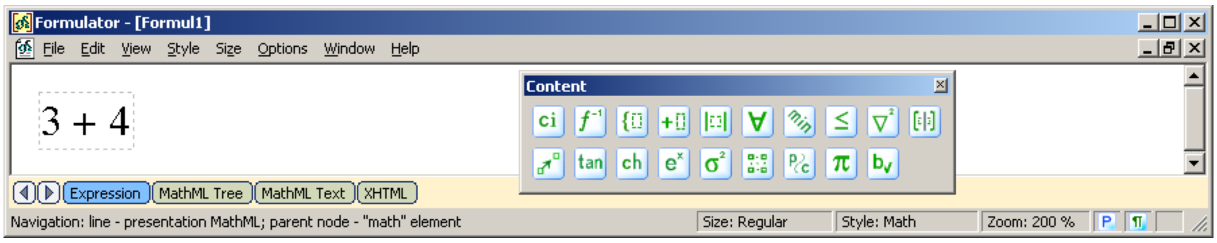
Entity name: cn

display

defines rendering in a display style or an in-line style


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Change Layout
Toolbars
Tabbed toolbar
Status Bar
Zoom
Show Nesting Ctrl+Shift+N
Show Read-Only Ctrl+Shift+R
Line Spacing...
Refresh All Through MathML
Content

Size: Regular Style: Math Zoom: 200 %


Formulator - [Formul1]
File Edit View Style Size Options Window Help
3 + 4
Content
Expression MathML Tree MathML Text XHTML

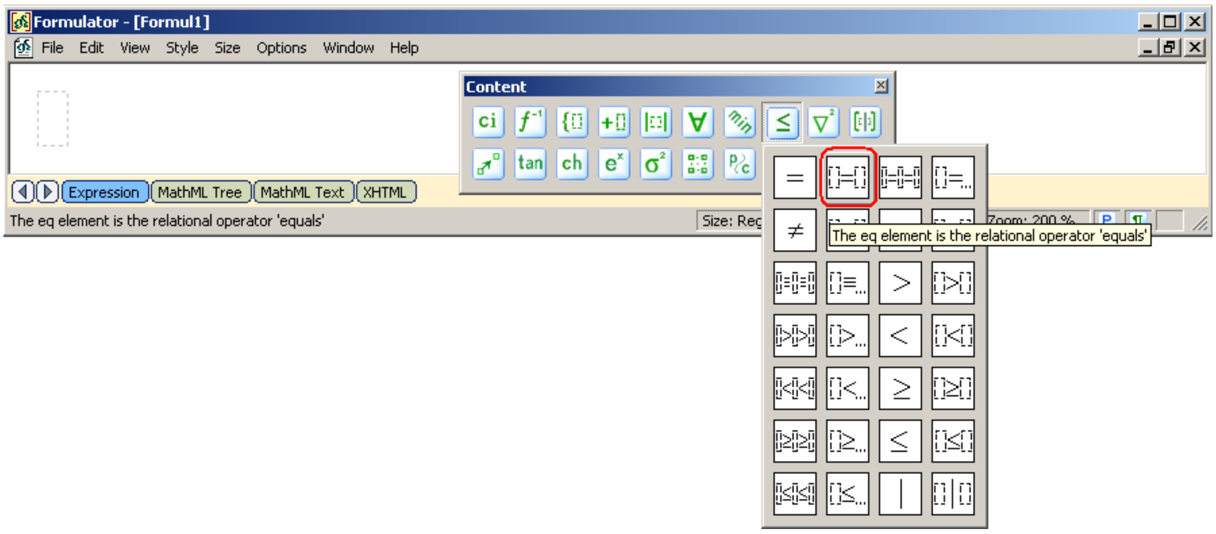
Navigation: line - presentation MathML; parent node - "math" element

Size: Regular Style: Math Zoom: 200 %

Basic Content Elements: functions and operators

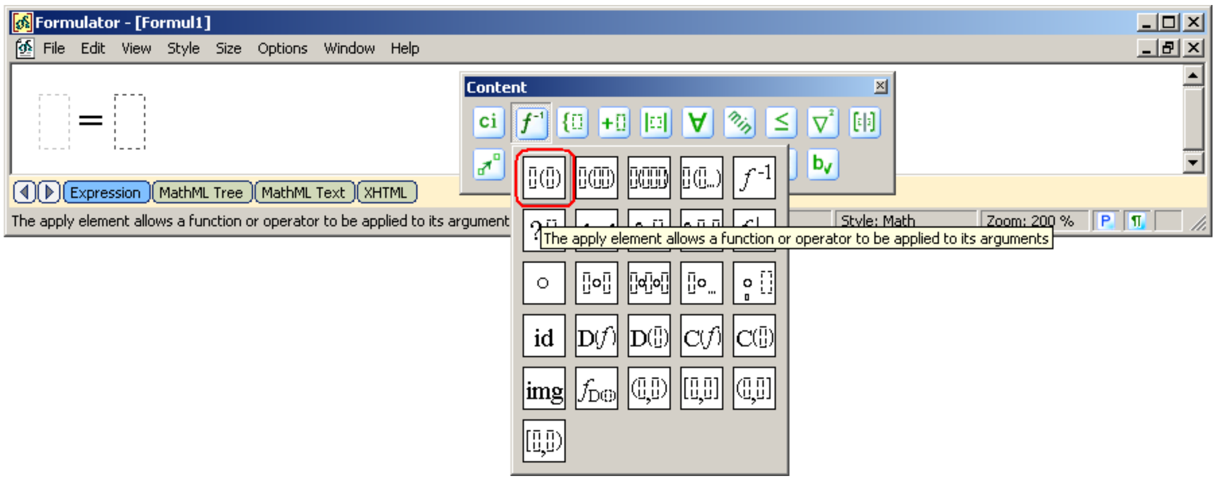
1. Create the image of a given function, which is the set of values taken by the function (4.4.2.14.2 example from the W3C MathML Recommendation):

```
<apply>
  <eq/>
  <apply><image/>
    <sin/>
  </apply>
  <interval>
    <cn>-1</cn>
    <cn> 1</cn>
  </interval>
</apply>
```



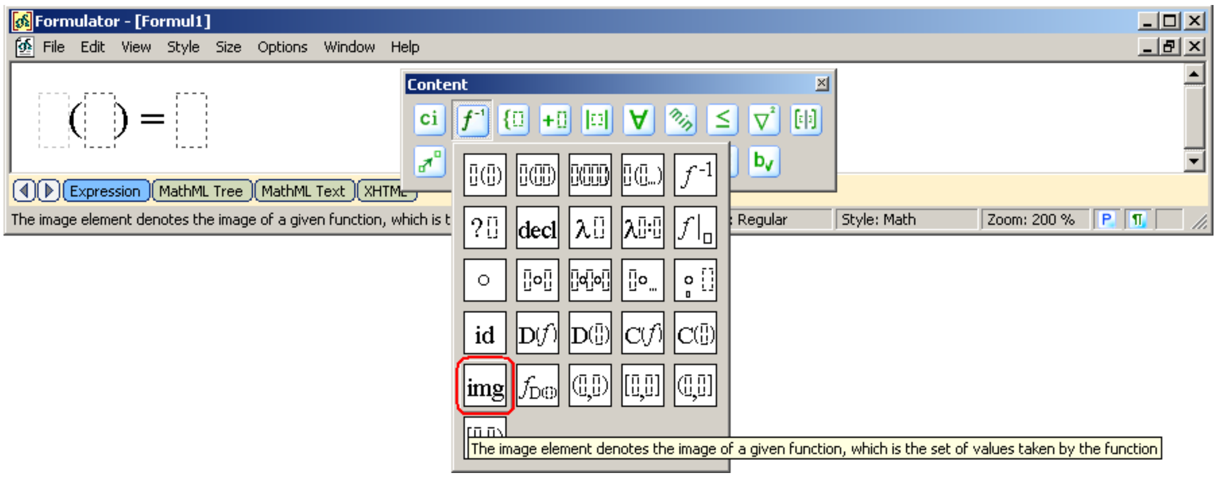
The eq element is the relational operator 'equals'

The eq element is the relational operator 'equals'



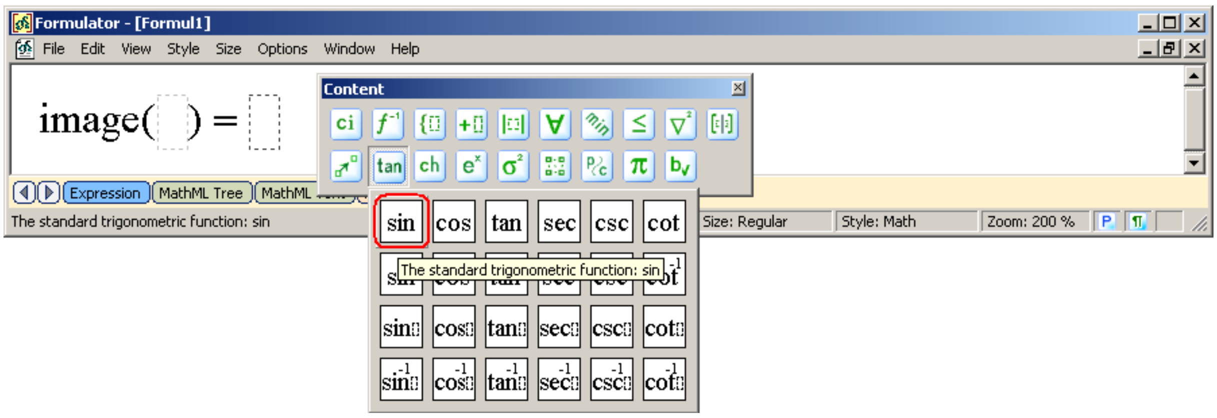
The apply element allows a function or operator to be applied to its argument

The apply element allows a function or operator to be applied to its arguments



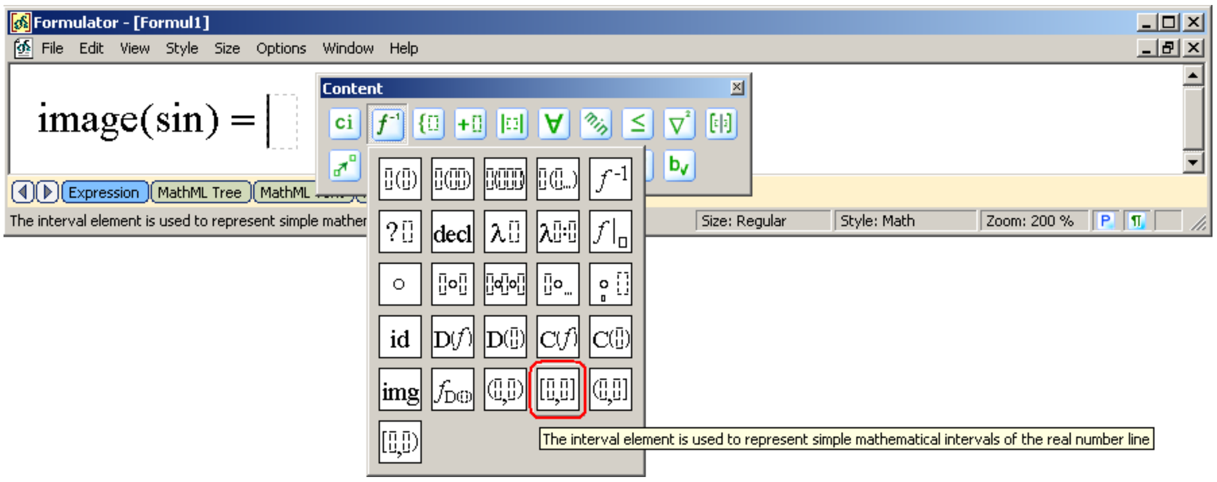
The image element denotes the image of a given function, which is t

The image element denotes the image of a given function, which is the set of values taken by the function


Formulator - [Formul1]

The standard trigonometric function: sin

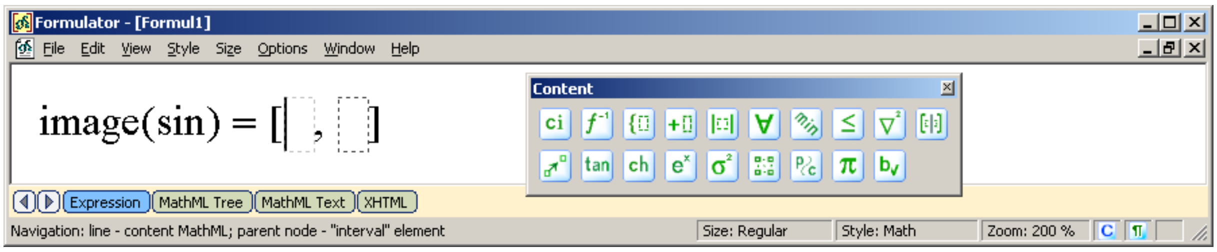
Size: Regular | Style: Math | Zoom: 200 %


Formulator - [Formul1]

The interval element is used to represent simple mather

The interval element is used to represent simple mathematical intervals of the real number line

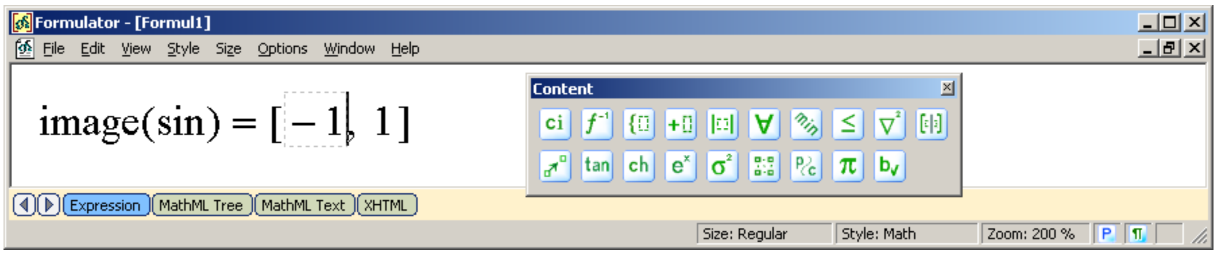
Size: Regular | Style: Math | Zoom: 200 %


Formulator - [Formul1]

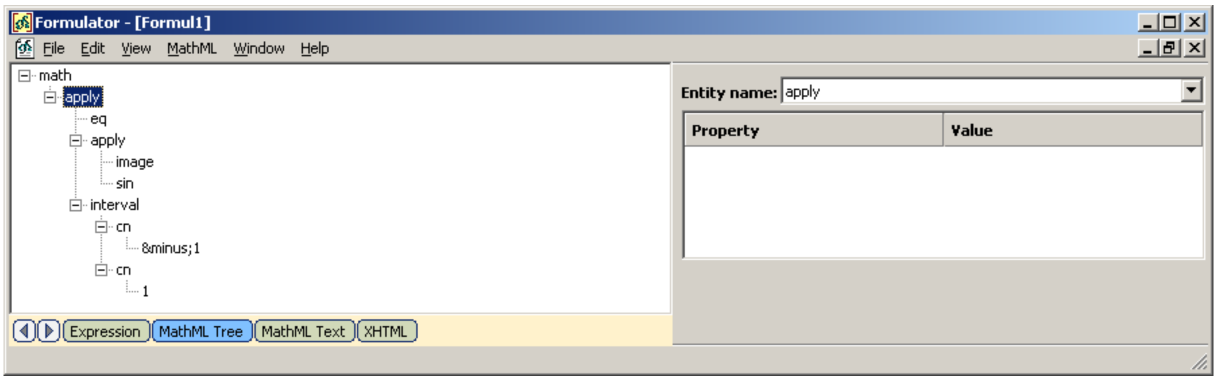
Navigation: line - content MathML; parent node - "interval" element

Size: Regular | Style: Math | Zoom: 200 %

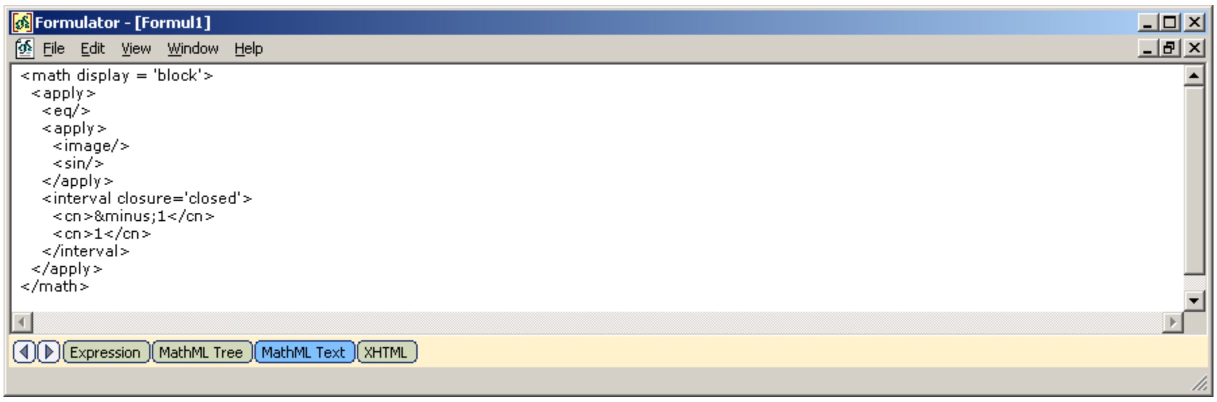
There comes a time when we should recall the difference between Presentation and Content markups. If we just type any text into the first and the second slot, MathML Weaver consider this as if we want to create some presentation for arguments of the “interval” element. But it is not true, since we are going to create content markup number elements there. So, now be careful, MathML Weaver will correctly detect the element of the content markup only if we type into these slots valid numeric constants.


Formulator - [Formul1]

Size: Regular | Style: Math | Zoom: 200 %


Formulator - [Formul1]
File Edit View MathML Window Help
Entity name: apply
Property
Value
Expression MathML Tree MathML Text XHTML

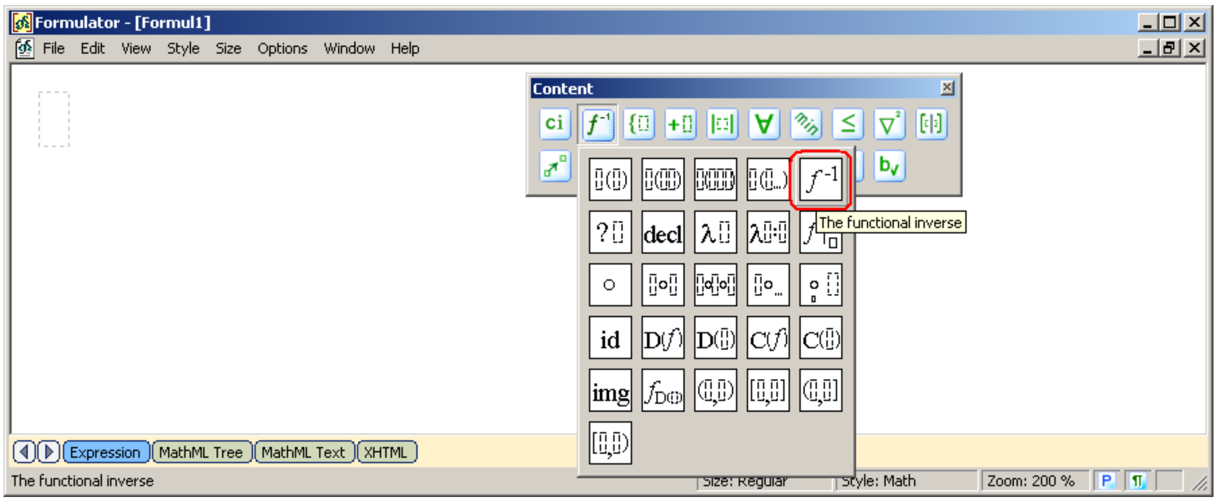
```
math
  apply
    eq
    apply
      image
      sin
    interval
      cn
        &minus;1
      cn
        1
```

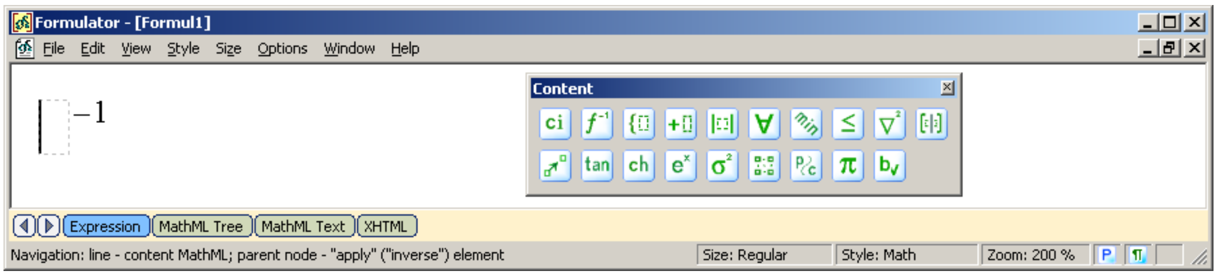

Formulator - [Formul1]
File Edit View Window Help
Expression MathML Tree MathML Text XHTML

```
<math display = 'block'>
 <apply>
  <eq/>
  <apply>
   <image/>
   <sin/>
  </apply>
  <interval closure='closed'>
   <cn>&minus;1</cn>
   <cn>1</cn>
  </interval>
 </apply>
</math>
```

2. Create the inverse element in order to construct a generic expression for the functional inverse of that function (4.4.2.5.2 example from the W3C MathML Recommendation):

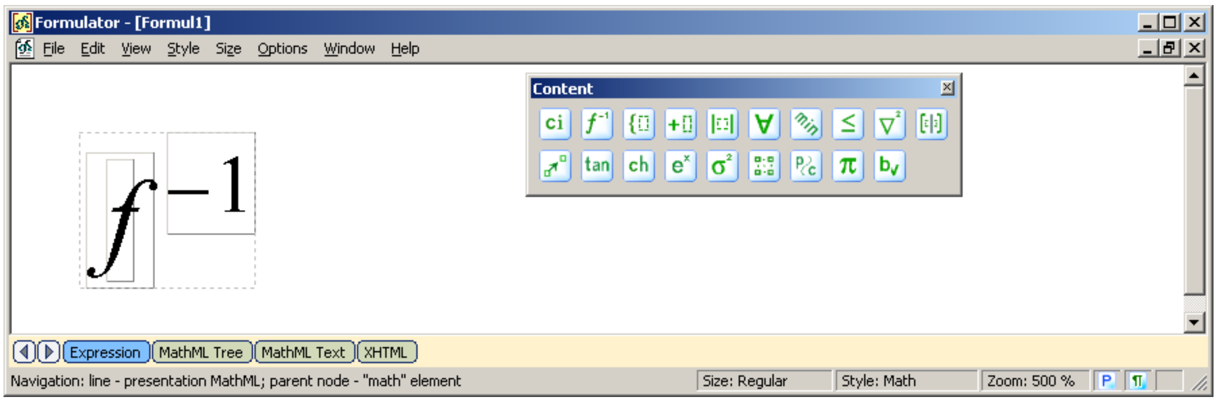
```
<apply>
  <inverse/>
  <ci> f </ci>
</apply>
```




Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci
tan
ch
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

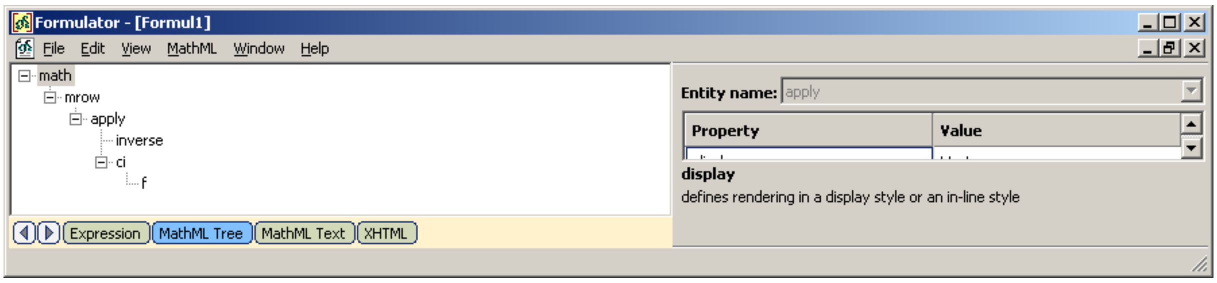
Navigation: line - content MathML; parent node - "apply" ("inverse") element

Type "f" into the input slot and see the structure of the MathML tree using nesting feature (Ctrl+Shift+N) and larger zoom (Ctrl+5).


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci tan ch π
Expression MathML Tree MathML Text XHTML

Navigation: line - presentation MathML; parent node - "math" element
Size: Regular
Style: Math
Zoom: 500 %

See results on the "MathML Tree" page.


Formulator - [Formul1]
File Edit View MathML Window Help
Expression MathML Tree MathML Text XHTML

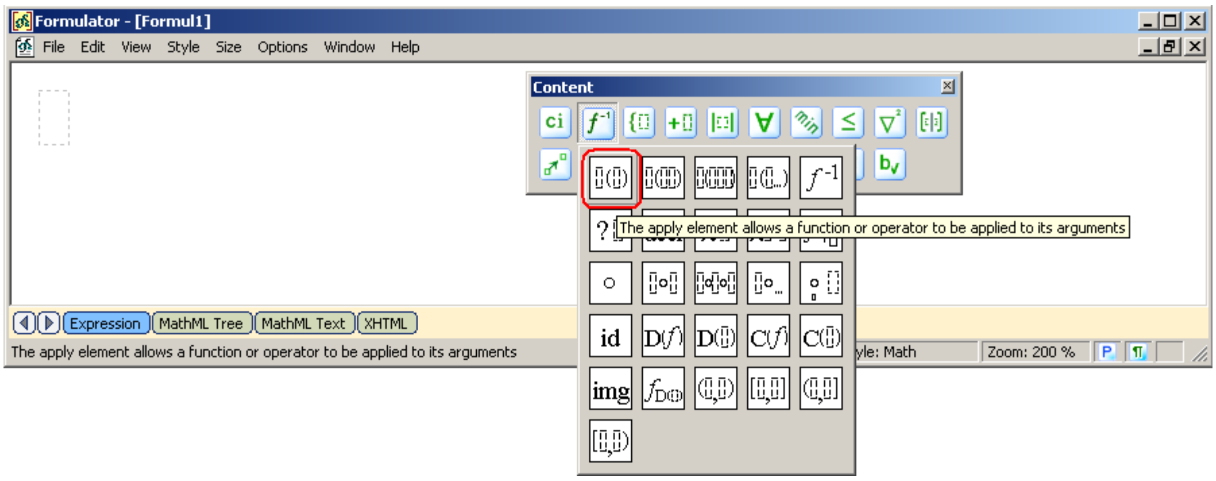
```
math
  mrow
    apply
      inverse
      ci
        f
```

Entity name: apply

Property	Value

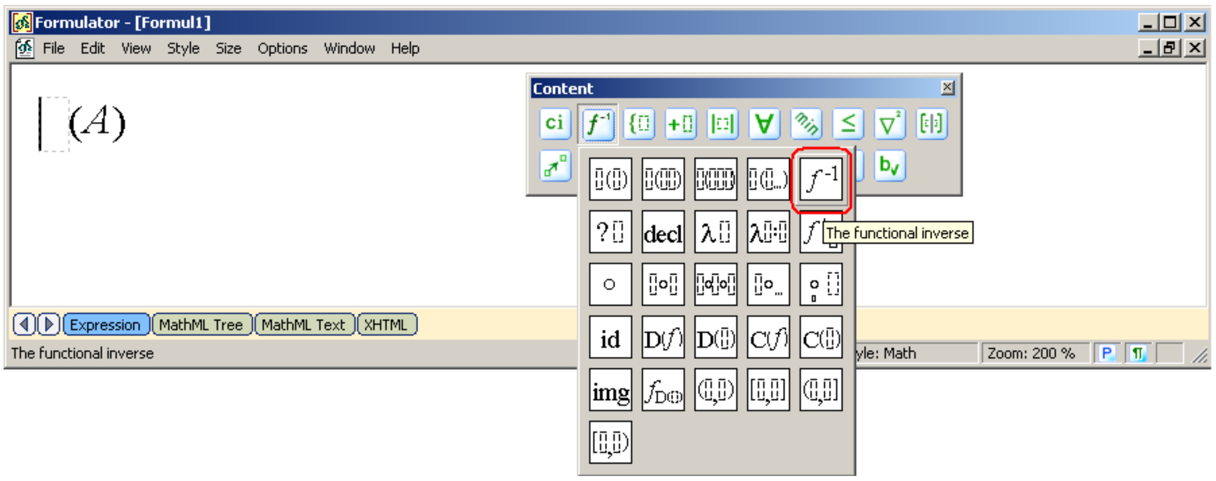
display
defines rendering in a display style or an in-line style

```
<apply>
  <apply><inverse/>
    <ci type="matrix"> a </ci>
  </apply>
  <ci> A </ci>
</apply>
```


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci
id
img
Expression MathML Tree MathML Text XHTML
yle: Math
Zoom: 200 %

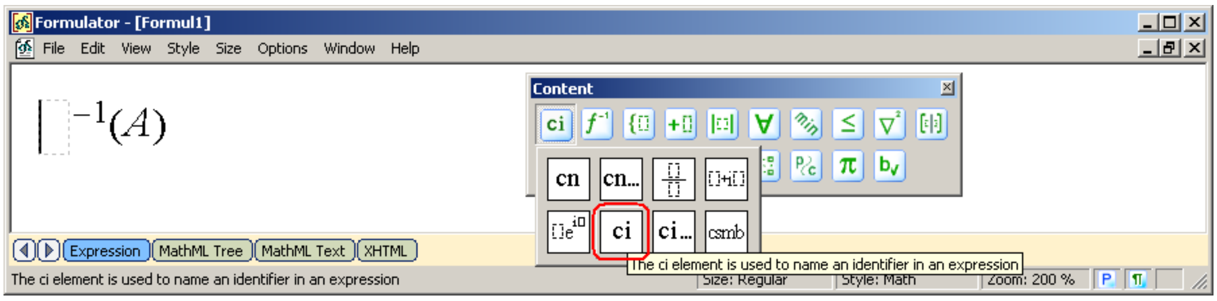
The apply element allows a function or operator to be applied to its arguments

The apply element allows a function or operator to be applied to its arguments



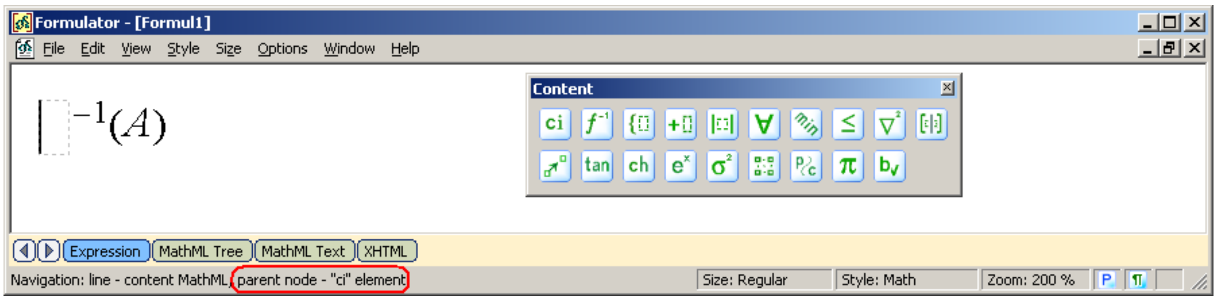
The functional inverse

Style: Math | Zoom: 200 %



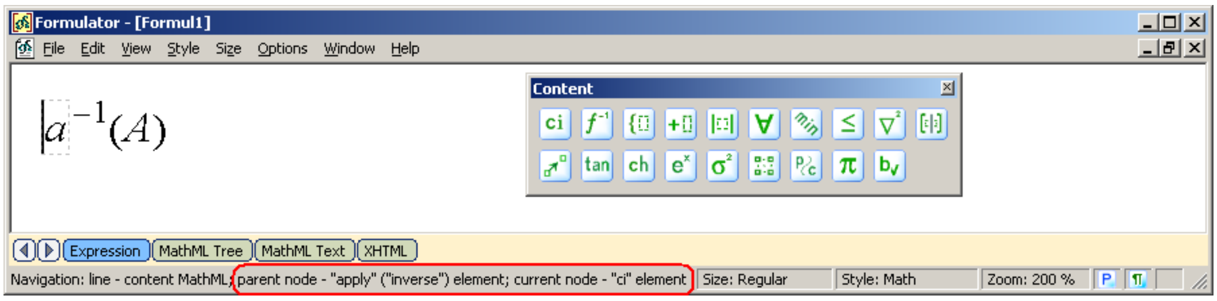
The ci element is used to name an identifier in an expression

Size: Regular | Style: Math | Zoom: 200 %



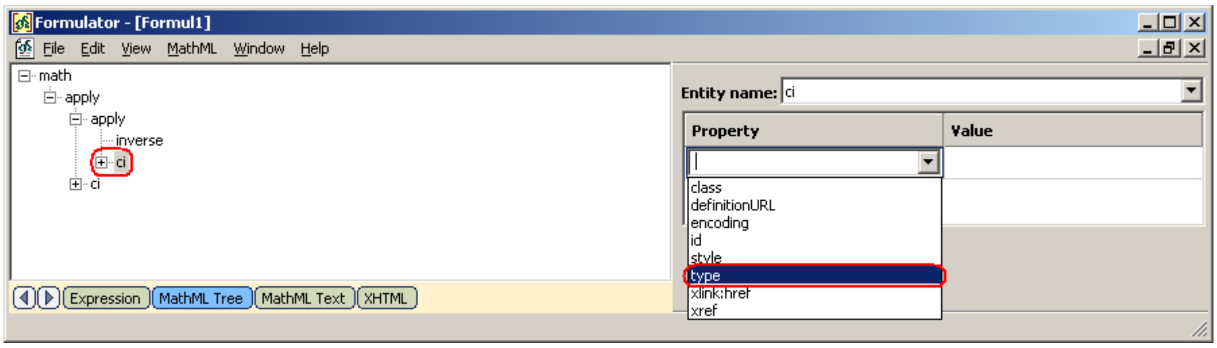
Navigation: line - content MathML; parent node - "ci" element

Size: Regular | Style: Math | Zoom: 200 %



Navigation: line - content MathML; parent node - "apply" ("inverse") element; current node - "ci" element

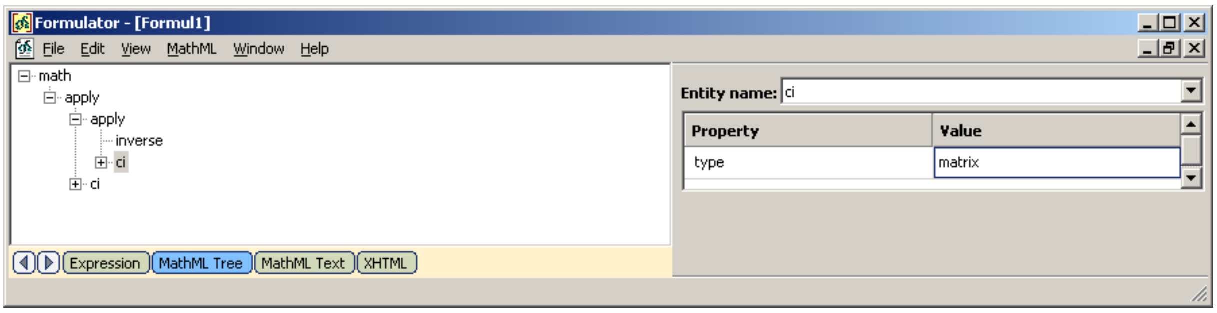
Size: Regular | Style: Math | Zoom: 200 %



```
math
  apply
    apply
      inverse
      ci
    ci
```

Entity name: ci

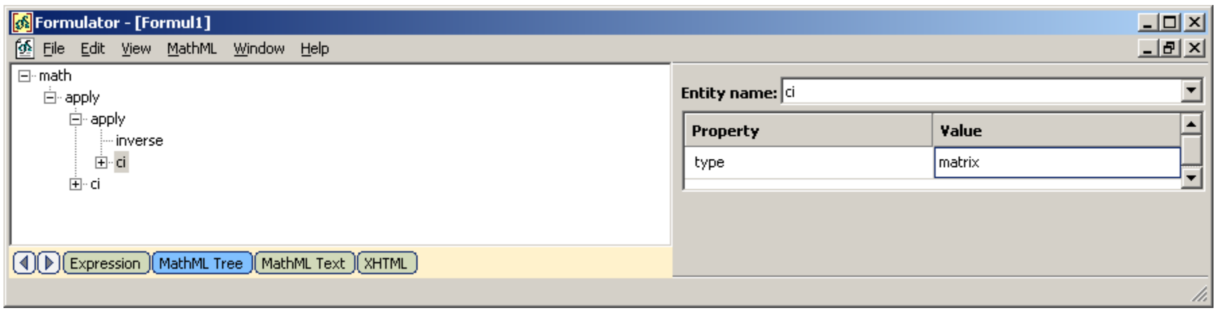
Property	Value
class	
definitionURL	
encoding	
id	
style	
type	
xlink:href	
xref	


Formulator - [Formul1]
File Edit View MathML Window Help
Expression MathML Tree MathML Text XHTML

```
math
  apply
    apply
      inverse
      ci
    ci
```

Entity name: ci

Property	Value
type	matrix


Formulator - [Formul1]
File Edit View MathML Window Help
Expression MathML Tree MathML Text XHTML

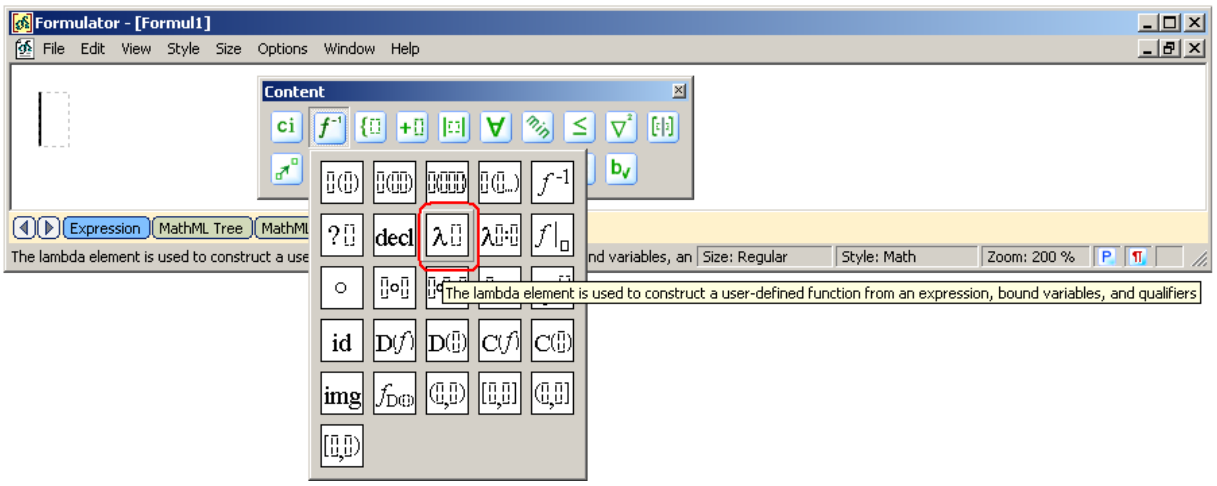
```
math
  apply
    apply
      inverse
      ci
    ci
```

Entity name: ci

Property	Value
type	matrix

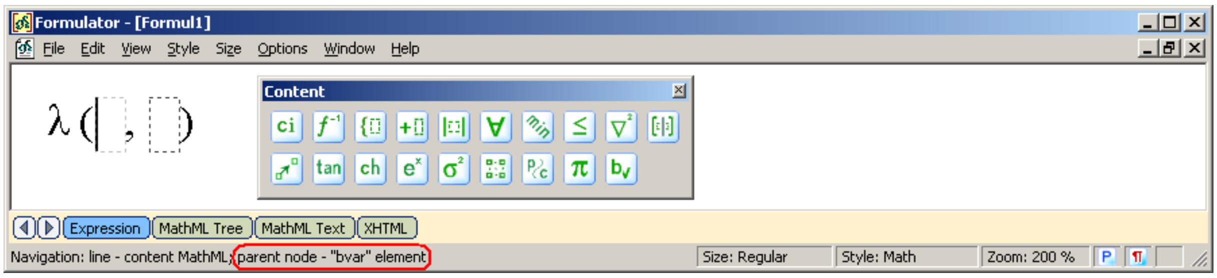
3. Create the lambda element that is used to construct a user-defined function from an expression, bound variables, and qualifiers (4.4.2.9.2 examples from the W3C MathML Recommendation):

```
<lambda>
  <bvar><ci> x </ci></bvar>
  <apply><sin/>
    <apply>
      <plus/>
      <ci> x </ci>
      <cn> 1 </cn>
    </apply>
  </apply>
</lambda>
```



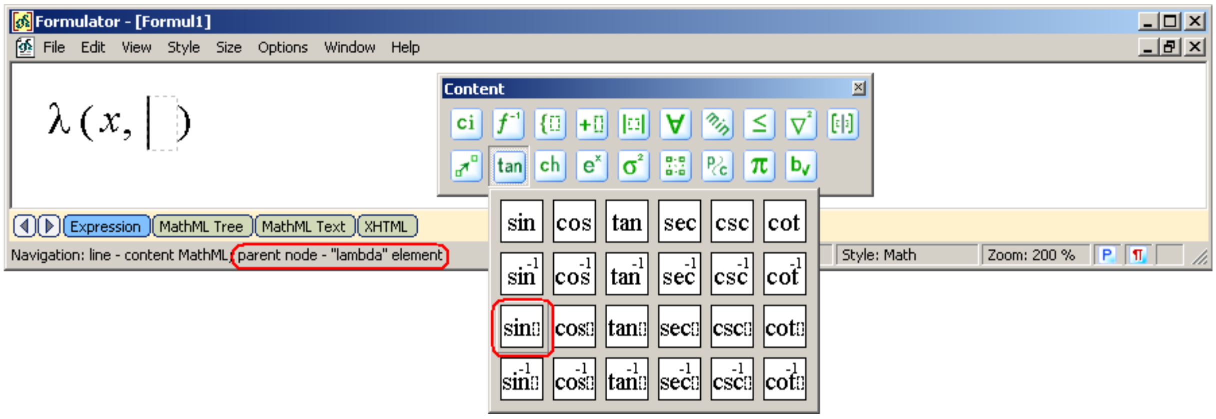
The lambda element is used to construct a user-defined function from an expression, bound variables, and qualifiers

The first input slot is for the “bvar” element; type “x” text into it and it will be detected as a “ci” element when converting to MathML.

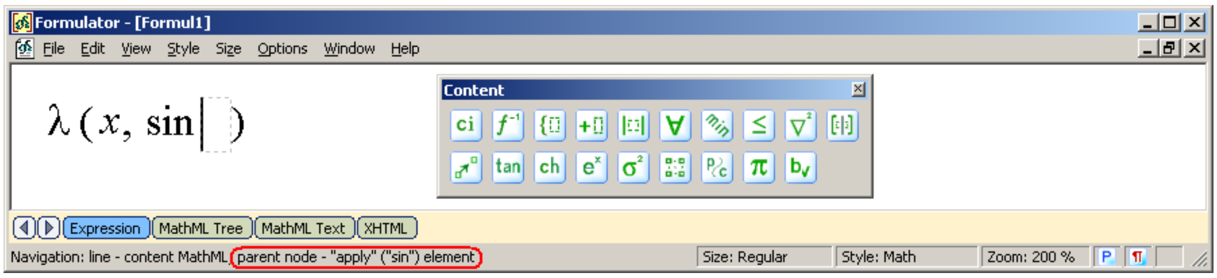

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

Navigation: line - content MathML; parent node - "bvar" element | Size: Regular | Style: Math | Zoom: 200 %

Now navigate to the second input slot (press the Right arrow) and insert a new visual form for editing sin(…) expression.

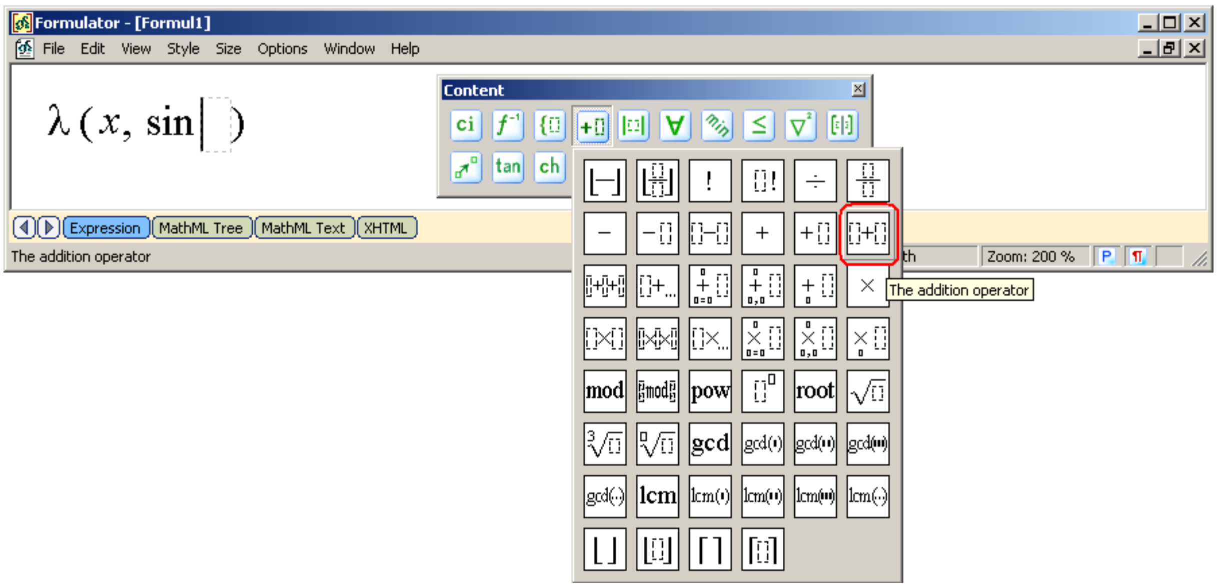

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
sin cos tan sec csc cot

Navigation: line - content MathML; parent node - "lambda" element | Style: Math | Zoom: 200 %

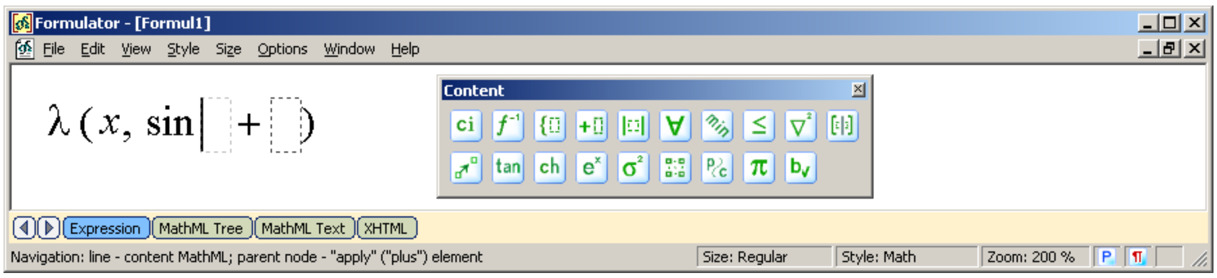

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

Navigation: line - content MathML; parent node - "apply" ("sin") element | Size: Regular | Style: Math | Zoom: 200 %

Now insert the “apply” element for the <plus/> operator. Please note that we cannot just type ‘+’ sign, because the current input mode is Presentation MathML (icon P in the right side of the Status Bar).


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
mod pow root gcd lcm
The addition operator

The addition operator | Zoom: 200 %

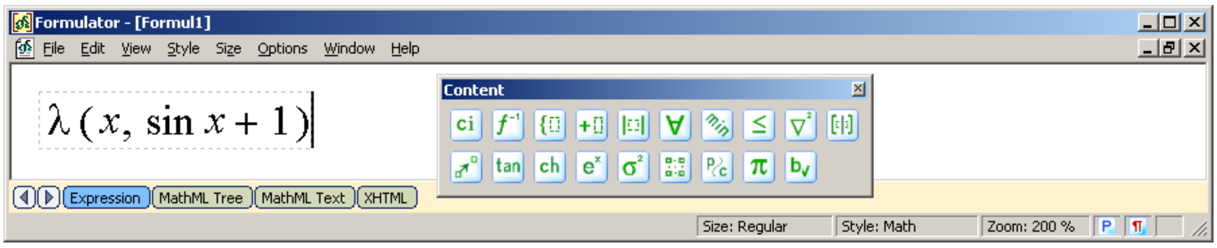

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

$\lambda\,(x,\ \sin\square + \square)$

Navigation: line - content MathML; parent node - "apply" ("plus") element

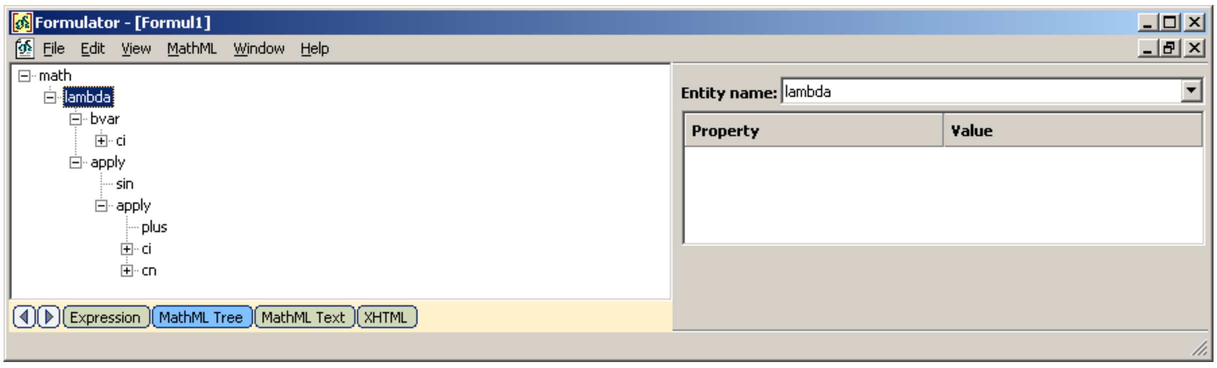
Size: Regular | Style: Math | Zoom: 200 %

Type "x" and "1" into input slots. The results are shown on the next two figures.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

$\lambda\,(x,\ \sin x + 1)$

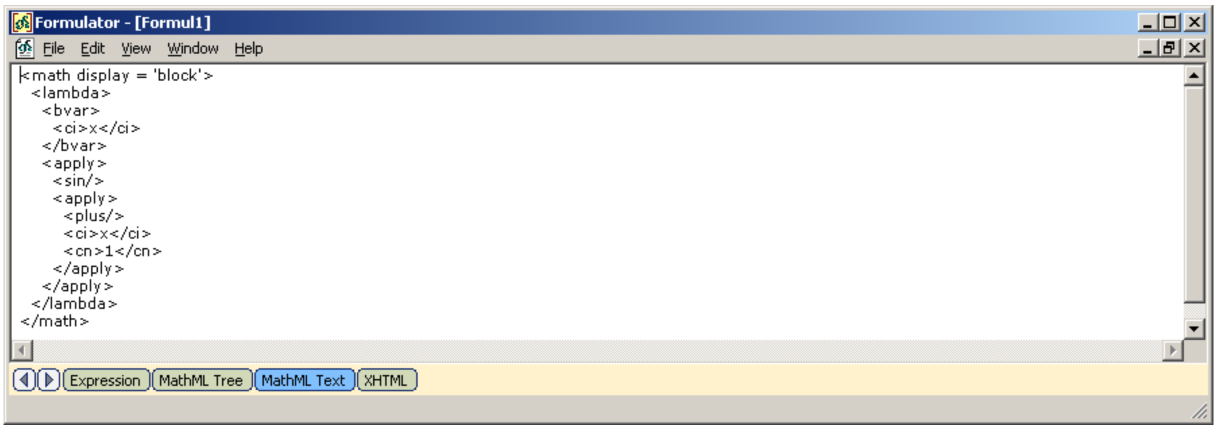
Size: Regular | Style: Math | Zoom: 200 %


Formulator - [Formul1]
File Edit View MathML Window Help
Expression MathML Tree MathML Text XHTML

```
math
  lambda
    bvar
      ci
    apply
      sin
      apply
        plus
        ci
        cn
```

Entity name: lambda

Property	Value


Formulator - [Formul1]
File Edit View Window Help
Expression MathML Tree MathML Text XHTML

```
<math display = 'block'>
  <lambda>
    <bvar>
      <ci>x</ci>
    </bvar>
    <apply>
      <sin/>
      <apply>
        <plus/>
        <ci>x</ci>
        <cn>1</cn>
      </apply>
    </apply>
  </lambda>
</math>
```

Basic Content Elements: invisible and transparent elements

This section describes how to work with invisible and transparent MathML nodes: "declare" and "condition" elements of the content markup.

Working with the "declare" element asks for additional actions, because according to the W3C MathML Recommendation it should not be directly rendered. Thus the "declare" element is considered as *invisible* and is rendered in two cases only:

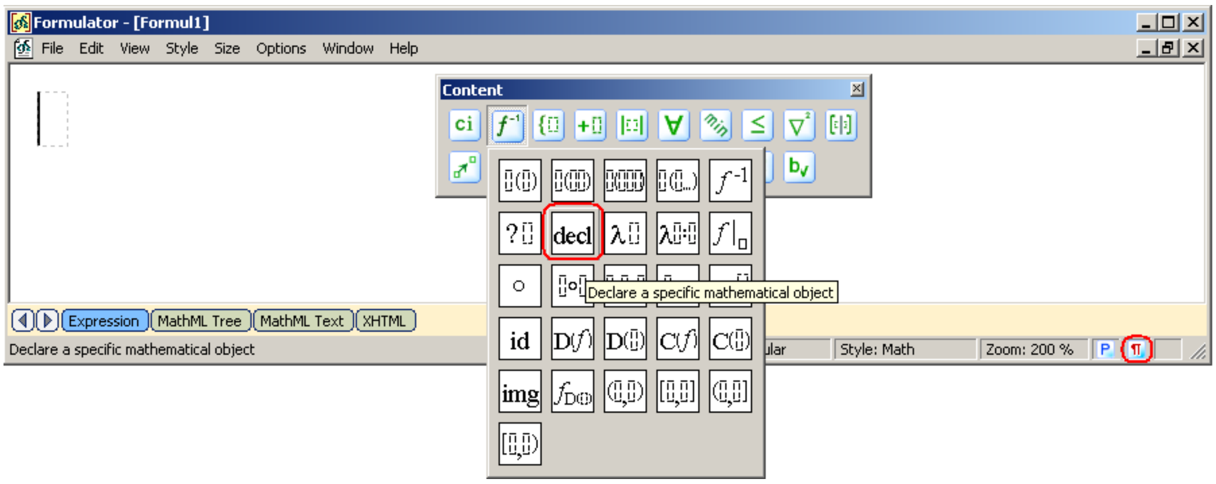
a) just after insertion of this element from the mathematical toolbar of MathML Weaver (f^{-1});

b) if the option "show invisible elements" is turned on (the green icon ¶ in the right corner of the Status Bar).

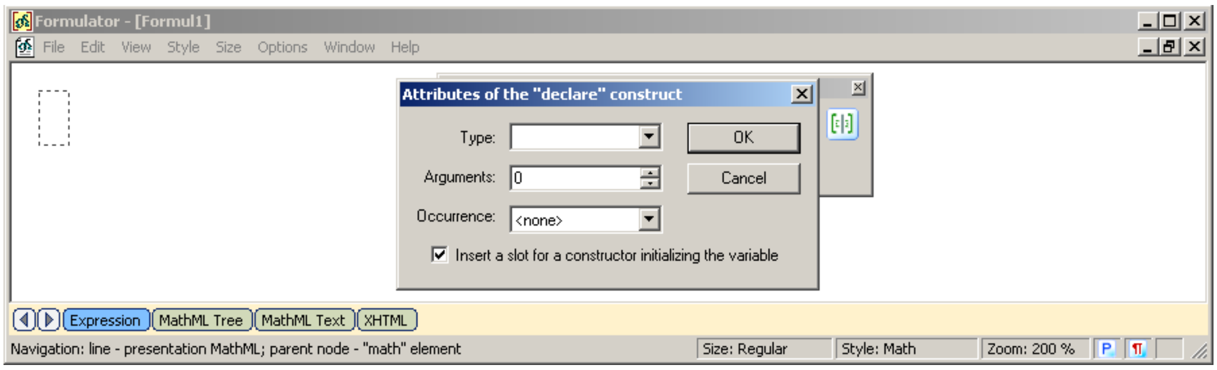
Additionally there is a limitation while using the "declare" element: MathML Weaver cannot hold a document that has no other elements except of the "declare". This means that when a user tries to save the following example without any additional MathML elements, then an empty document will be created. This known limitation of MathML Weaver relates to the specific rendering requirements of the "declare" element and the point that the presence of this element makes a sense only when other elements of the document refers to this declaration.

1. In order to demonstrate how to work with the "declare" element let's create a MathML tree for the following example from the W3C MathML Recommendation (4.4.2.8.1)

```
<declare type="vector">
  <ci> V </ci>
  <vector>
    <cn> 1 </cn><cn> 2 </cn><cn> 3 </cn>
  </vector>
</declare>
```


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci
decl
Declare a specific mathematical object
Expression MathML Tree MathML Text XHTML
Declare a specific mathematical object
Style: Math
Zoom: 200 %

Please note that currently rendering of invisible elements is turned off (¶).


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Expression MathML Tree MathML Text XHTML

Attributes of the "declare" construct

Type:

Arguments: 0

Occurrence: <none>

☑ Insert a slot for a constructor initializing the variable

OK

Cancel

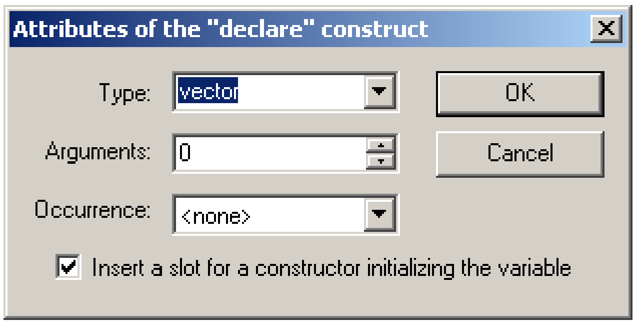
Navigation: line - presentation MathML; parent node - "math" element

Size: Regular

Style: Math

Zoom: 200 %

A dialog is shown to initially tune attributes of the "declare" element. Select the "vector" in the "Type" drop-down list. The next two attributes ("nargs", "occurrence") can be leave as they are, without editing. The check-box to the bottom of the dialog asks whether the "declare" element contains a constructor initializing the variable.

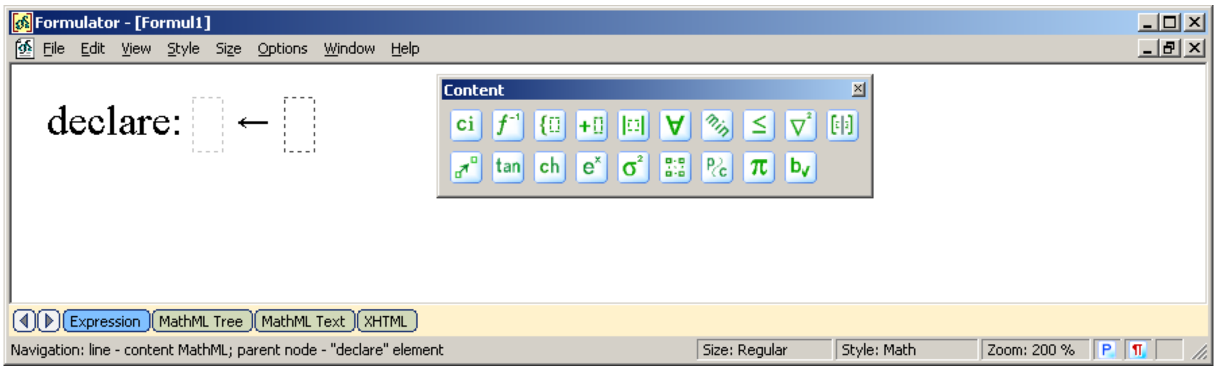

Attributes of the "declare" construct

Type: vector

Arguments: 0

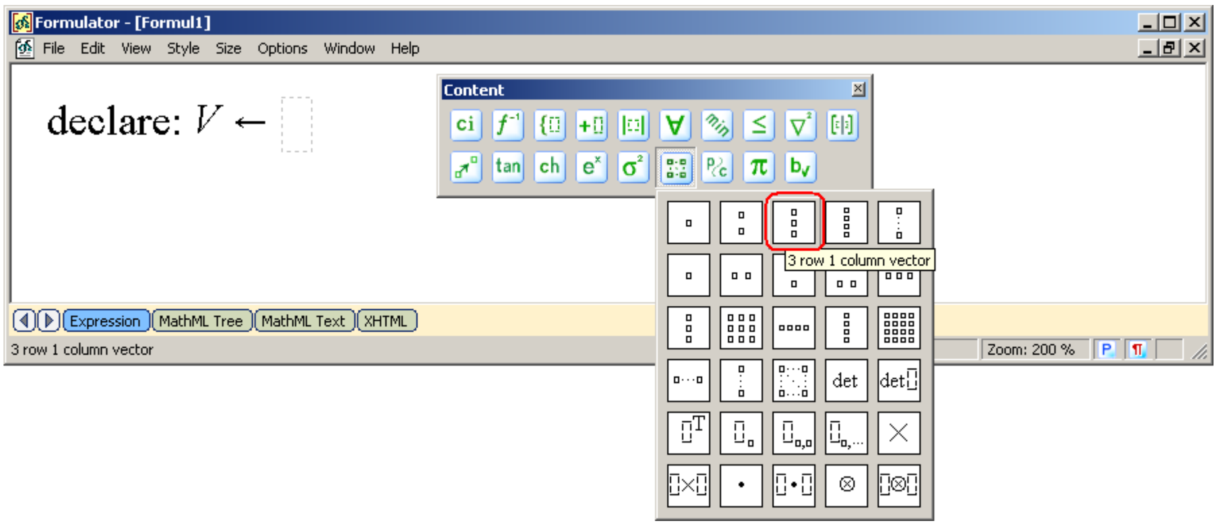
Occurrence: <none>

[✓] Insert a slot for a constructor initializing the variable

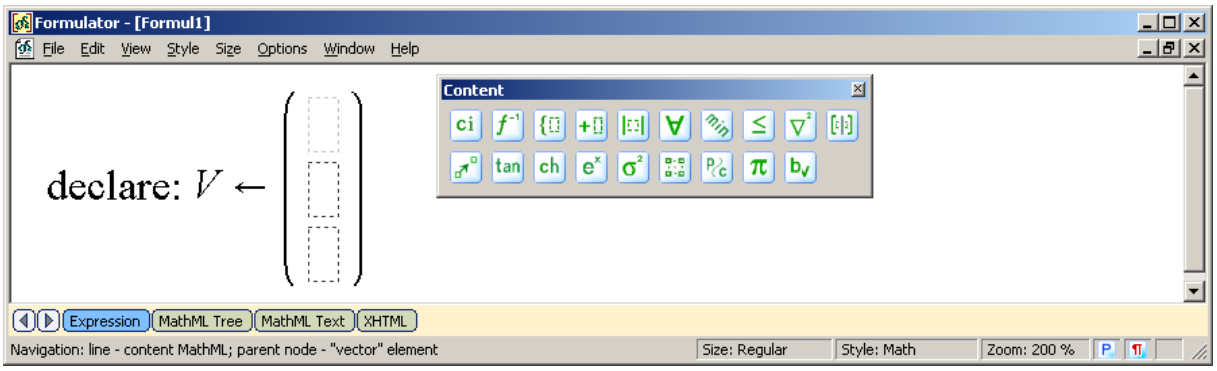

Formulator - [Formul1]
File Edit View Style Size Options Window Help
declare: ←
Content
Expression MathML Tree MathML Text XHTML

Navigation: line - content MathML; parent node - "declare" element Size: Regular Style: Math Zoom: 200 %

Type "V" into the first input slot; place the cursor into the second input slot and press the button for the "vector" element insertion.

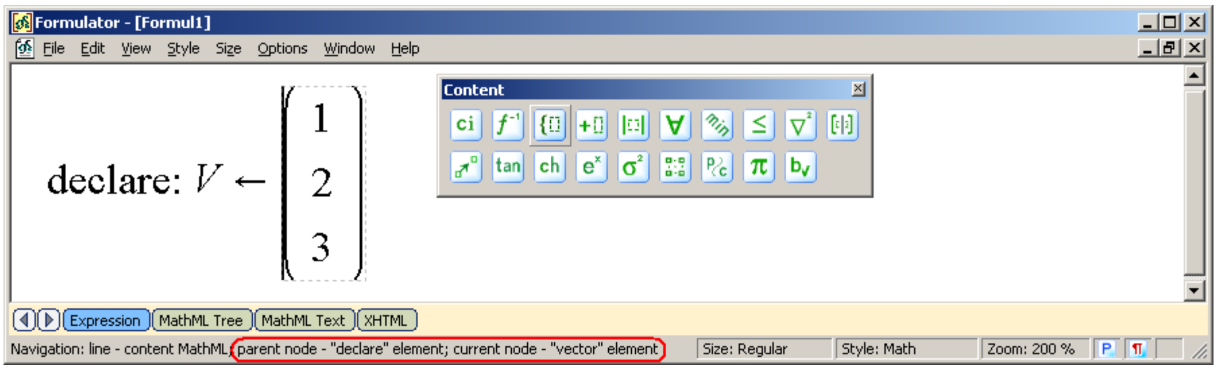

Formulator - [Formul1]
File Edit View Style Size Options Window Help
declare: V ←
Content
3 row 1 column vector
Expression MathML Tree MathML Text XHTML

3 row 1 column vector Zoom: 200 %


Formulator - [Formul1]
File Edit View Style Size Options Window Help
declare: V ←
Content
Expression MathML Tree MathML Text XHTML

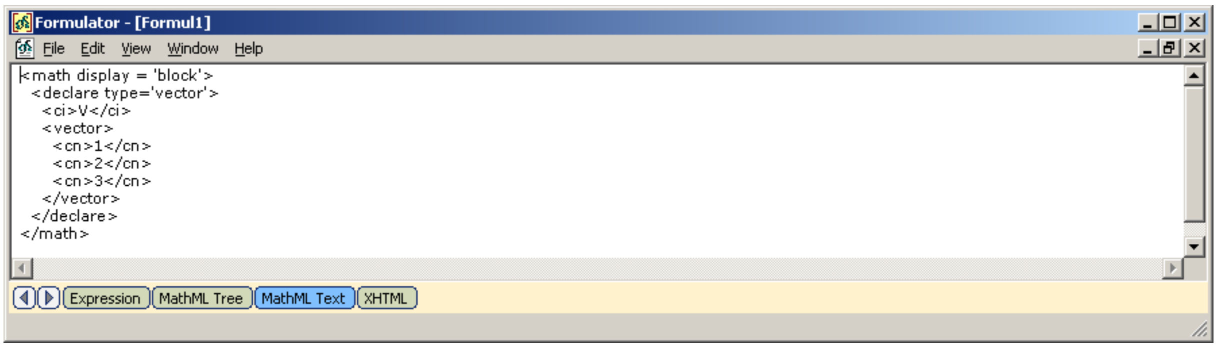
Navigation: line - content MathML; parent node - "vector" element Size: Regular Style: Math Zoom: 200 %

Type values of the vector into three input slots.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
declare: $V \leftarrow \begin{pmatrix}1\\2\\3\end{pmatrix}$
Expression MathML Tree MathML Text XHTML

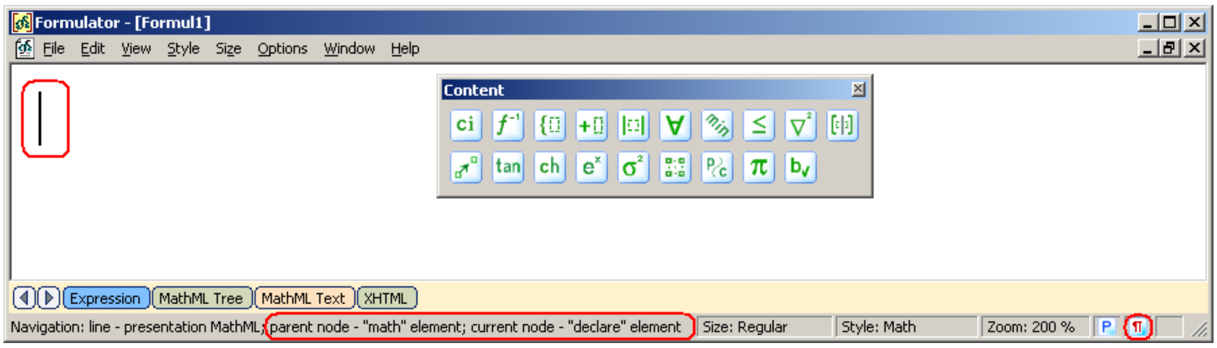
Navigation: line - content MathML; parent node - "declare" element; current node - "vector" element | Size: Regular | Style: Math | Zoom: 200 %

See the results.


Formulator - [Formul1]
File Edit View Window Help
Expression MathML Tree MathML Text XHTML

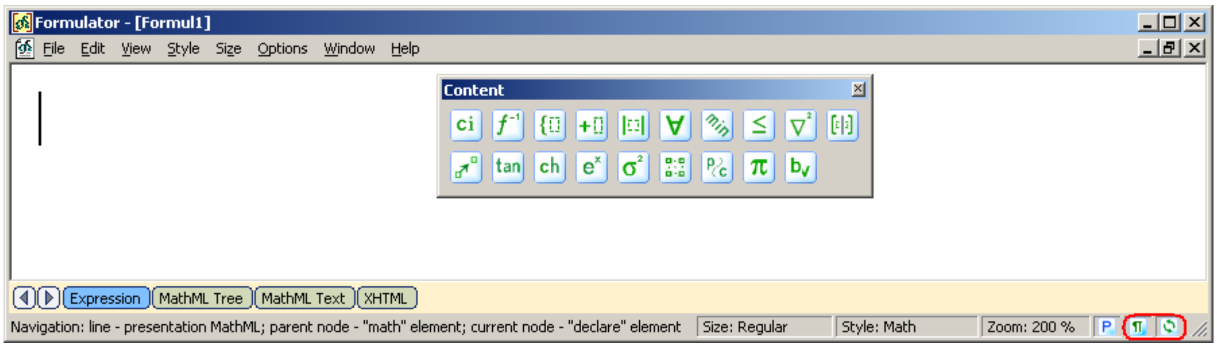
```
<math display = 'block'>
  <declare type='vector'>
    <ci>V</ci>
    <vector>
      <cn>1</cn>
      <cn>2</cn>
      <cn>3</cn>
    </vector>
  </declare>
</math>
```

Now do a inessential editing of the MathML text (for example, just press a space somewhere between tags) and get back to the “Expression” page. We see an empty body of the document, because the “declare” element is not rendered normally and the option “show invisible elements” is turned off currently. About presence of the just created “declare” element indicates only the navigation information bar and absence of the dashed line around an empty input slot (it is shown only when the line of a document contains no elements).

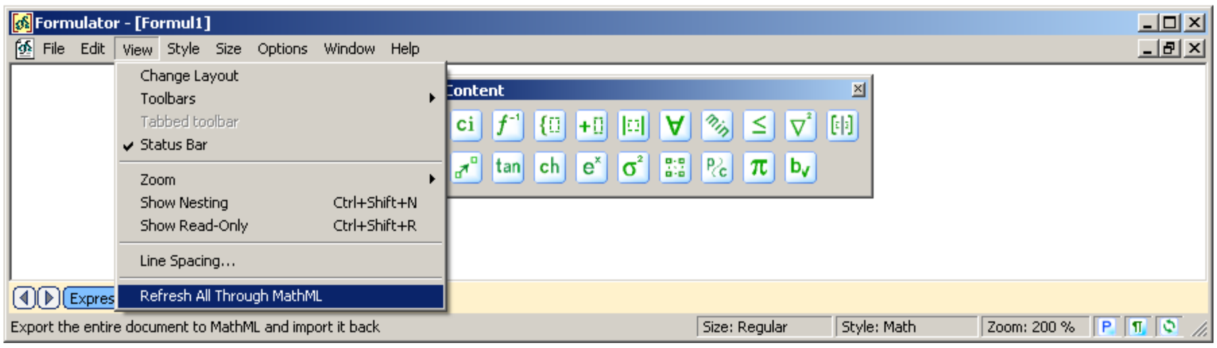

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

Navigation: line - presentation MathML; parent node - "math" element; current node - "declare" element | Size: Regular | Style: Math | Zoom: 200 %

Check the option “show invisible elements” from the “Display Content MathML Elements” submenu of the “Option” menu. We still can’t see the “declare” element, but the icon ¶ suggests that the action was successful and the icon prompts to the need of refreshing the document through MathML (the “Refresh All Through MathML” command from the “View” menu).

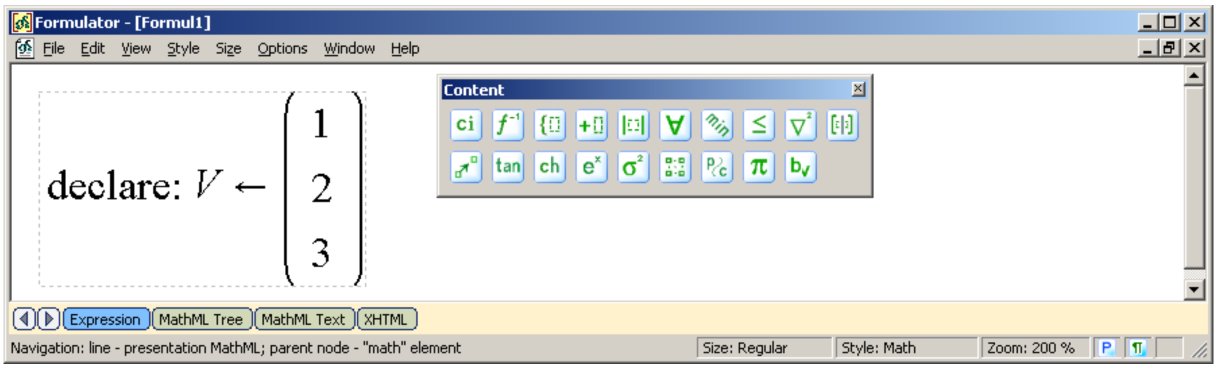

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

Navigation: line - presentation MathML; parent node - "math" element; current node - "declare" element | Size: Regular | Style: Math | Zoom: 200 %


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Change Layout
Toolbars
Tabbed toolbar
Status Bar
Zoom
Show Nesting Ctrl+Shift+N
Show Read-Only Ctrl+Shift+R
Line Spacing...
Refresh All Through MathML
Content

Export the entire document to MathML and import it back | Size: Regular | Style: Math | Zoom: 200 %

After refreshing the document rendering we see the “declare” element again, because the mode of invisible elements rendering is now turned on.

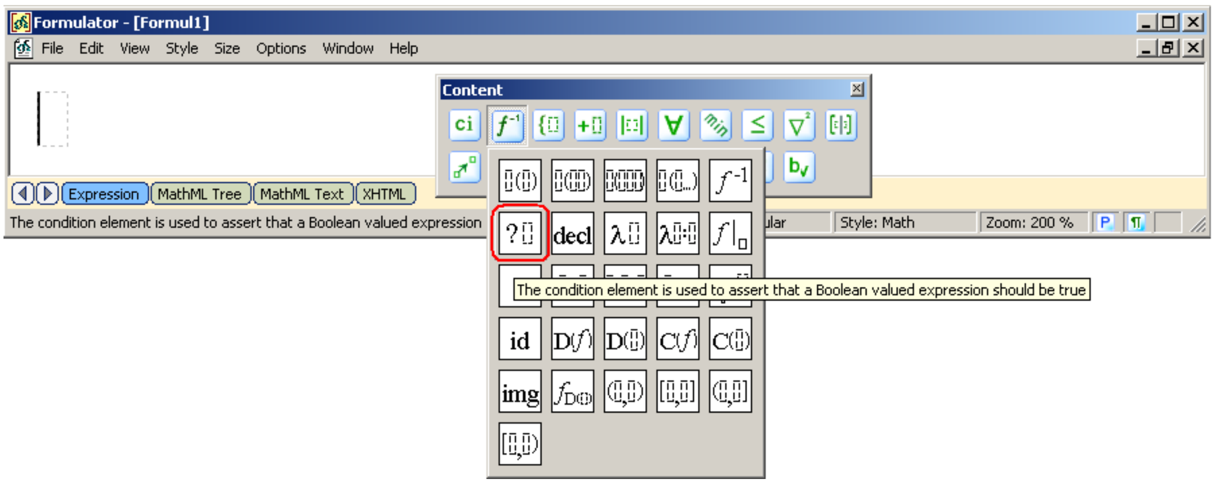

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

$$\text{declare: } V \leftarrow \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

Navigation: line - presentation MathML; parent node - "math" element | Size: Regular | Style: Math | Zoom: 200 %

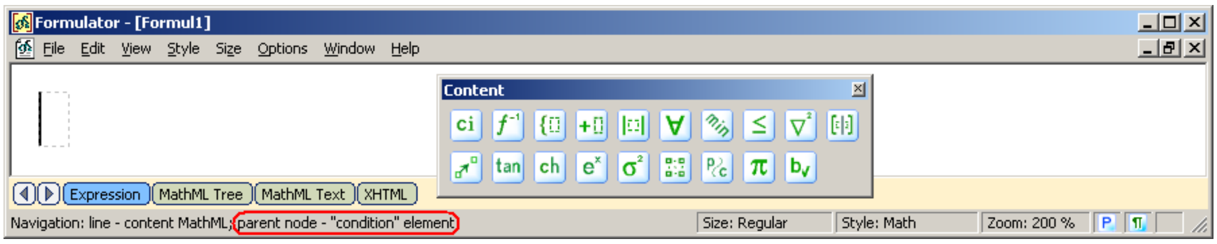
2. Create the “condition” element (4.4.2.7.2 examples from the W3C MathML Recommendation) to see how initially invisible (transparent) elements behave.

```
<condition>
  <apply><in/><ci> x </ci><ci type="set"> A </ci></apply>
</condition>
```


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Style: Math
Zoom: 200 %

The condition element is used to assert that a Boolean valued expression

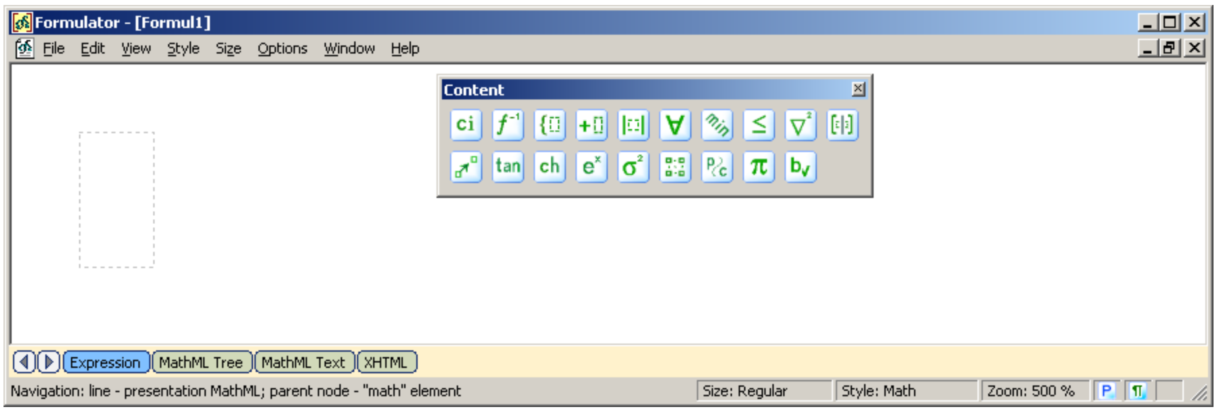
The condition element is used to assert that a Boolean valued expression should be true


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

Navigation: line - content MathML; parent node - "condition" element

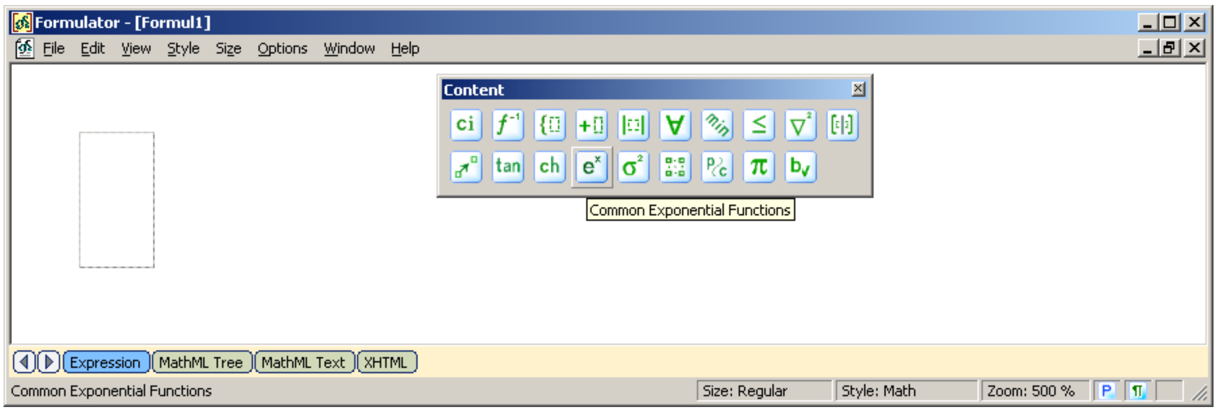
At the first glance, the rendering of the document is not changed, but the navigation information suggests that the "condition" element is successfully inserted and it is not rendered just because there is no presentation for it; the presentation of the "condition" element coincides with its contents rendering.

The changed structure of the document can be viewed using the "nesting view" feature. Compare view of the document before insertion of the "condition" element (dashed line around the empty input slot):


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 500 %

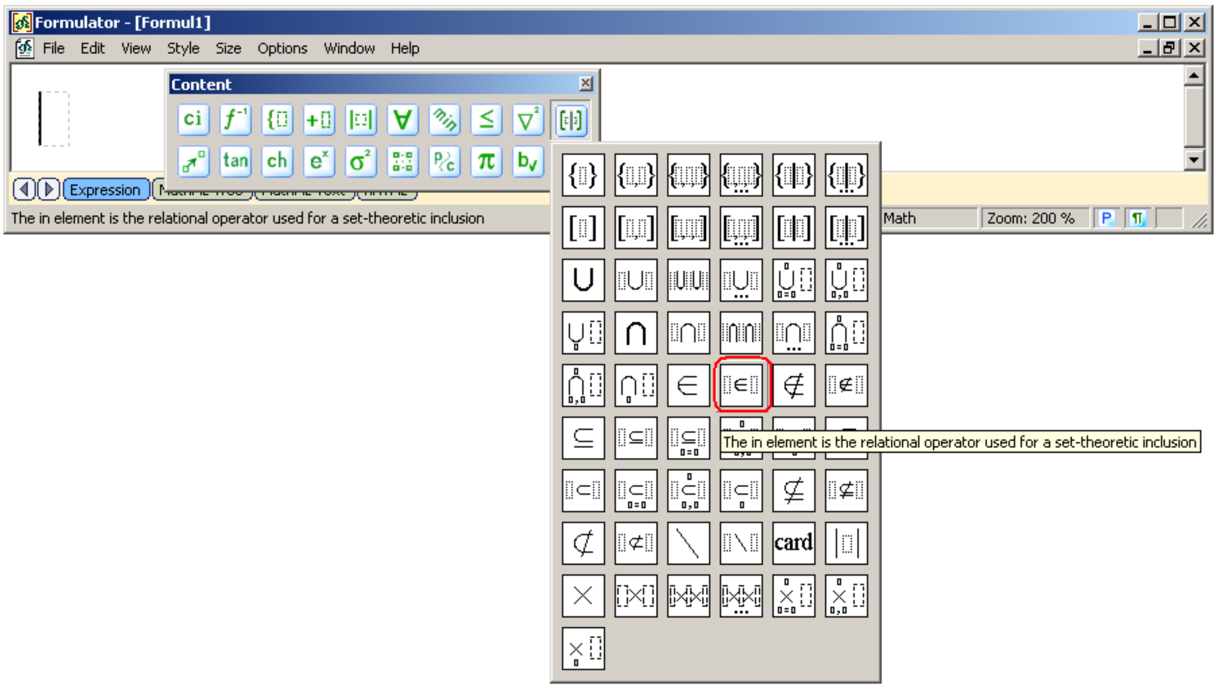
Navigation: line - presentation MathML; parent node - "math" element

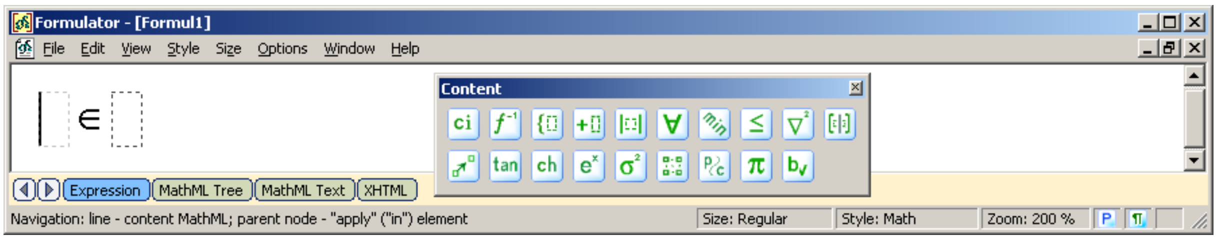
and after it (solid line over the dashed line around the empty input slot):

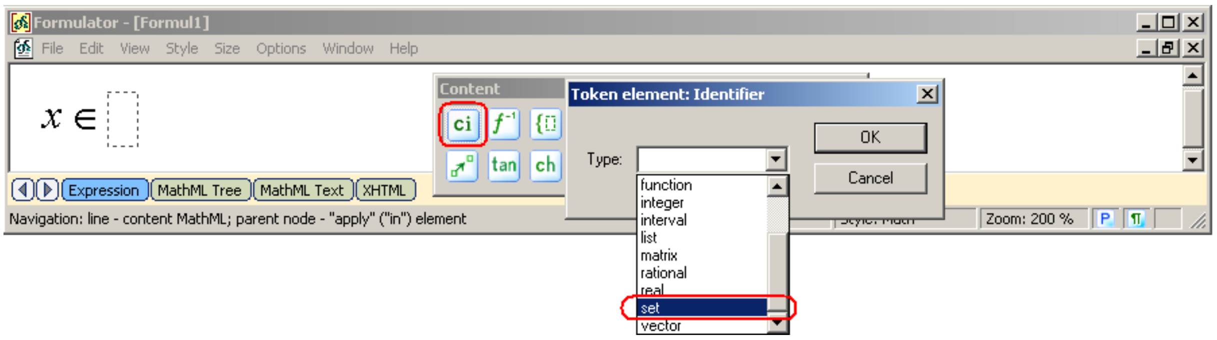


This solid line suggests to the thought that there is an internal frame element to hold a newly created element of the content markup.

Now let's back to our example and create the "apply" element inside of the "condition" element.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression
The in element is the relational operator used for a set-theoretic inclusion
Math
Zoom: 200 %
card
The in element is the relational operator used for a set-theoretic inclusion


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression
MathML Tree
MathML Text
XHTML
Navigation: line - content MathML; parent node - "apply" ("in") element
Size: Regular
Style: Math
Zoom: 200 %

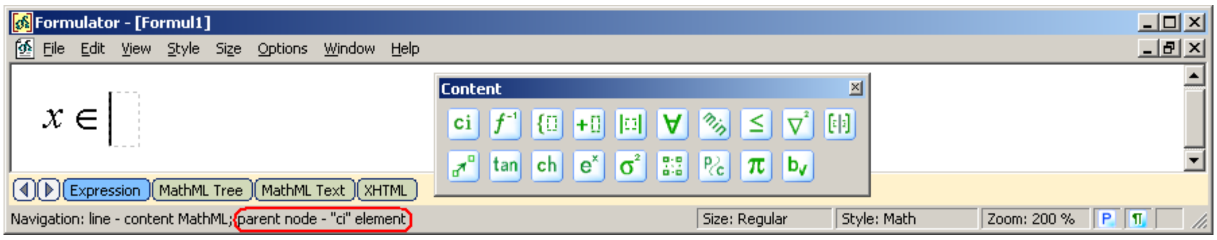

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci
tan
ch
Token element: Identifier
OK
Cancel
Expression MathML Tree MathML Text XHTML
Zoom: 200 %

$x \in$

Navigation: line - content MathML; parent node - "apply" ("in") element

Type:

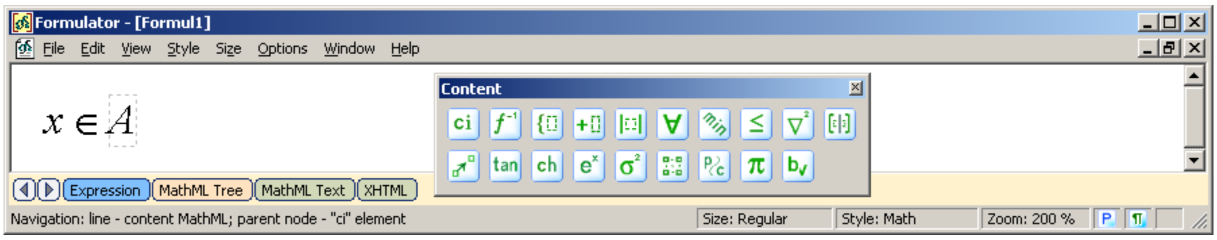
- function
- integer
- interval
- list
- matrix
- rational
- real
- [✓ set]
- vector


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci
tan
ch
Expression MathML Tree MathML Text XHTML

$x \in$

Navigation: line - content MathML; parent node - "ci" element

Size: Regular Style: Math Zoom: 200 %

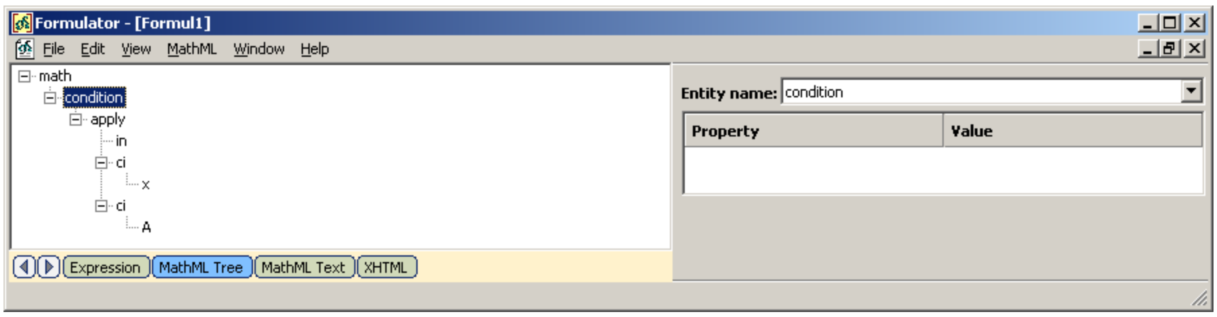

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci
tan
ch
Expression MathML Tree MathML Text XHTML

$x \in A$

Navigation: line - content MathML; parent node - "ci" element

Size: Regular Style: Math Zoom: 200 %

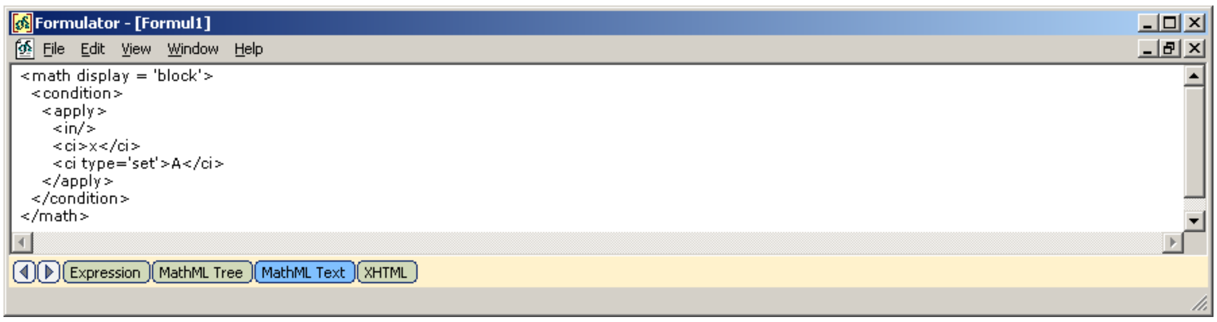
See the results


Formulator - [Formul1]
File Edit View MathML Window Help
Expression MathML Tree MathML Text XHTML

```
math
  condition
    apply
      in
      ci
        x
      ci
        A
```

Entity name: condition

Property	Value


Formulator - [Formul1]
File Edit View Window Help
Expression MathML Tree MathML Text XHTML

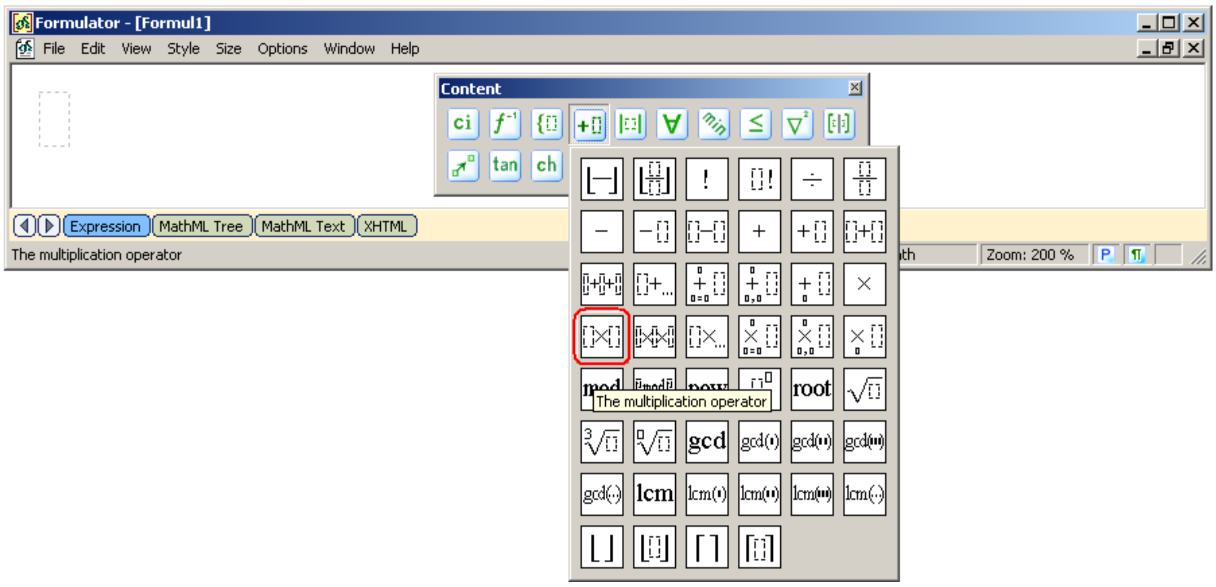
```
<math display = 'block'>
 <condition>
  <apply>
   <in/>
   <ci>x</ci>
   <ci type='set'>A</ci>
  </apply>
 </condition>
</math>
```

Arithmetic, Algebra, Logic and Relations

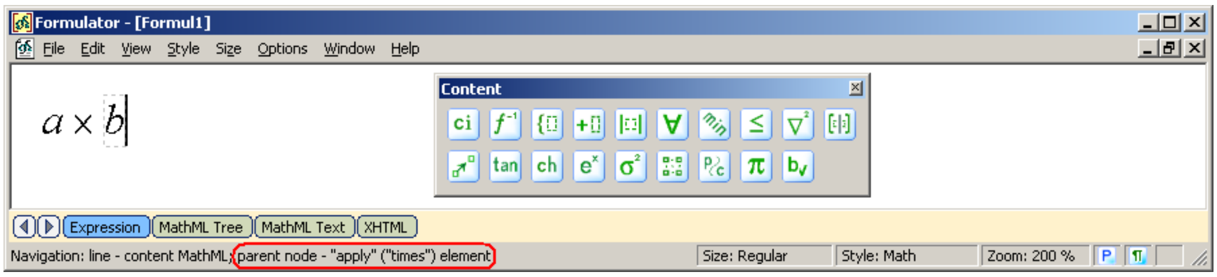
1. Example 4.4.3.9.2 from the W3C MathML Recommendation).

```
<apply>
  <times/>
  <ci> a </ci>
  <ci> b </ci>
</apply>
```

Press the button on the mathematical toolbar for arithmetic operators or switch the input mode to the Content Markup and just press the '*' character.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
The multiplication operator
Zoom: 200 %

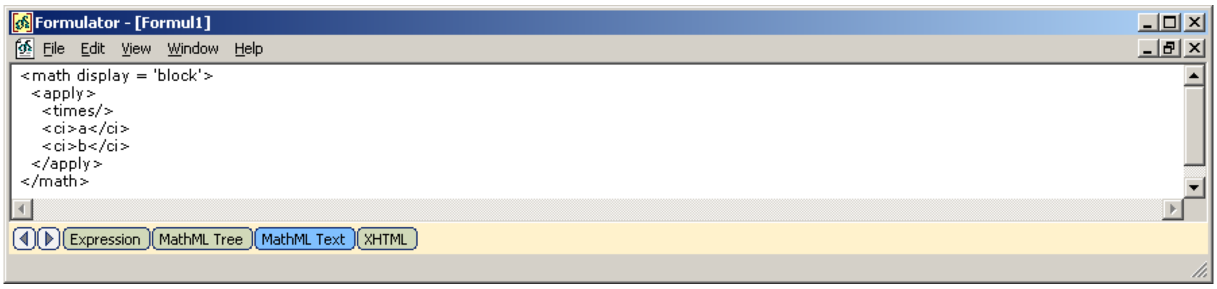
Type 'a', press the Right arrow, type 'b'.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

$a \times b$

Navigation: line - content MathML; parent node - "apply" ("times") element

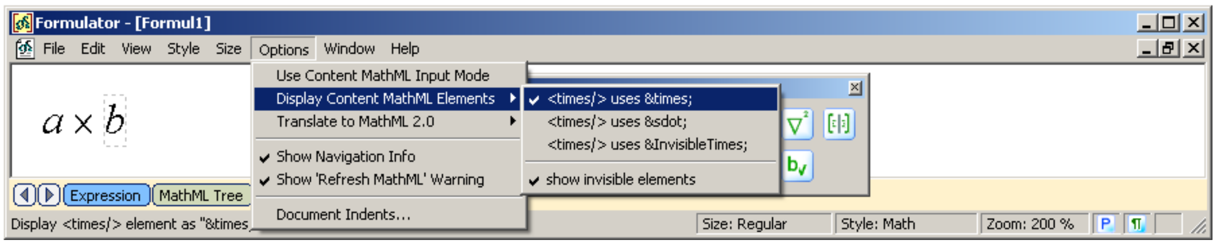
See the results


Formulator - [Formul1]
File Edit View Window Help
Expression MathML Tree MathML Text XHTML

```
<math display = 'block'>
 <apply>
  <times/>
  <ci>a</ci>
  <ci>b</ci>
 </apply>
</math>
```

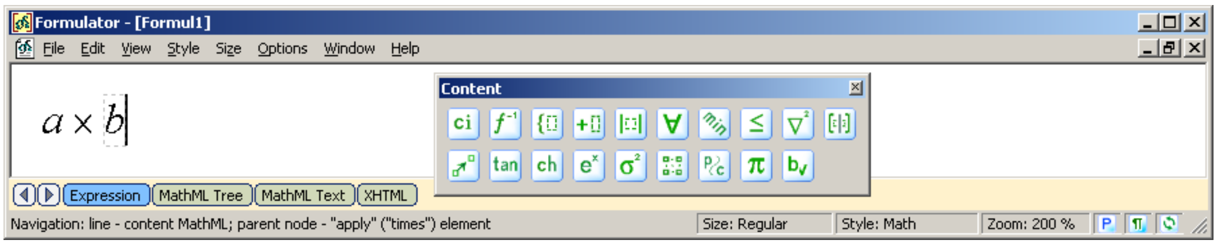
Now get back to the "Expression" page and see how to change the appearance of the <times/> element.

The next figures shows which item of menu controls the rendering option of the <times/> element.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Use Content MathML Input Mode
Display Content MathML Elements
Translate to MathML 2.0
Show Navigation Info
Show 'Refresh MathML' Warning
Document Indents...
<times/> uses &times;
<times/> uses &sdot;
<times/> uses &InvisibleTimes;
show invisible elements
Expression MathML Tree
Size: Regular Style: Math Zoom: 200 %

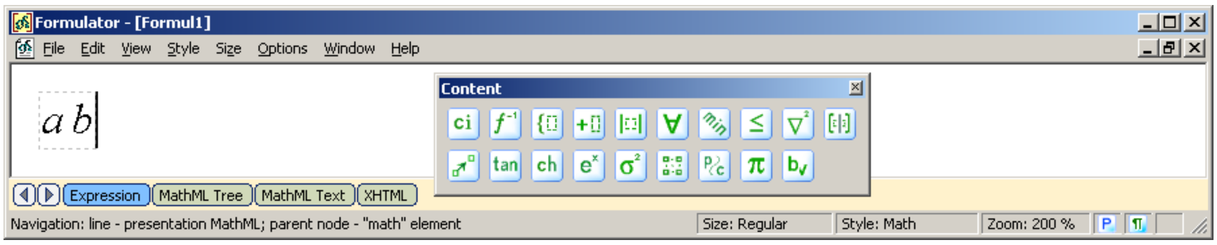
Display <times/> element as "&times

If we select another option ("&InvisibleTimes;") the appearance of the formula is not changed, but the icon prompts to the need of refreshing the document through MathML (the "Refresh All Through MathML" command from the "View" menu).


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci tan ch
Expression MathML Tree MathML Text XHTML
Size: Regular Style: Math Zoom: 200 %

Navigation: line - content MathML; parent node - "apply" ("times") element

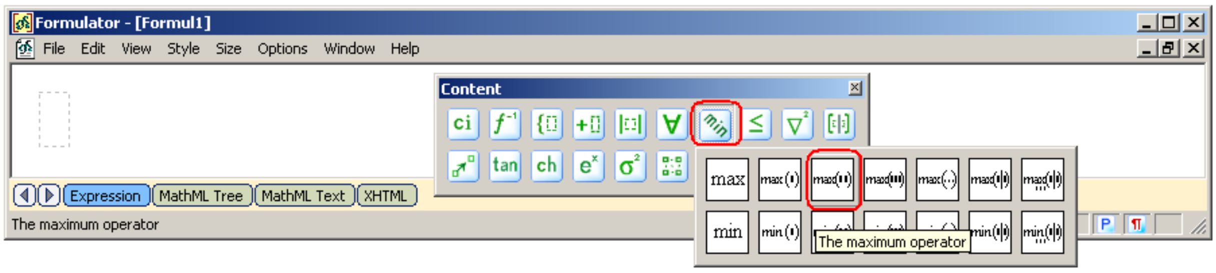
After refreshing the formula is rendered as if instead of the <times/> element is used the presentation element <mo>&InvisibleTimes</mo>.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci tan ch
Expression MathML Tree MathML Text XHTML
Size: Regular Style: Math Zoom: 200 %

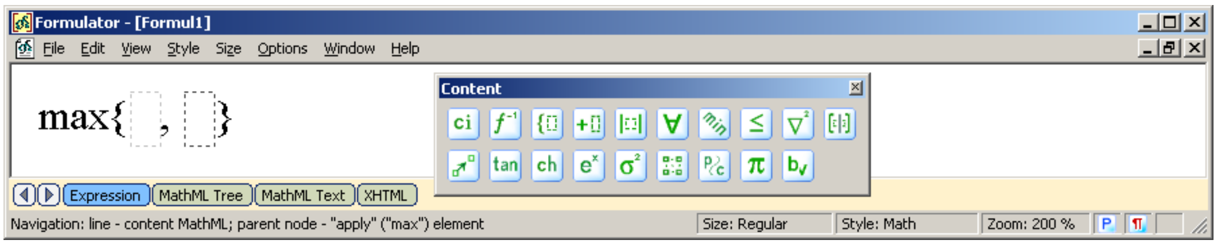
Navigation: line - presentation MathML; parent node - "math" element

2. Example 4.4.3.4.2 from the W3C MathML Recommendation): Maximum and minimum.

```
<apply>
  <max/>
  <ci> a </ci>
  <ci> b </ci>
</apply>
```

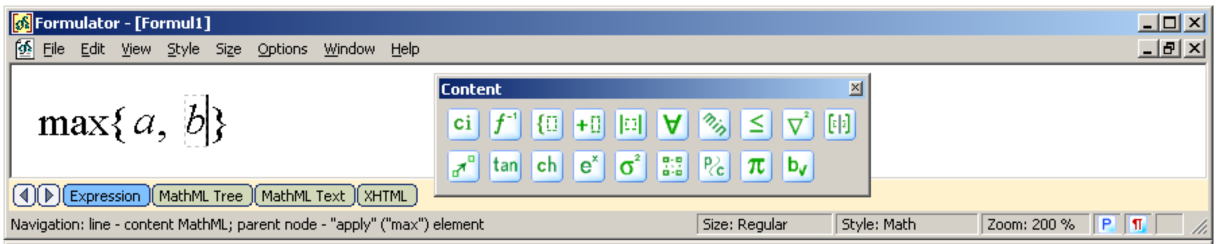

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci tan ch
max
min
The maximum operator
Expression MathML Tree MathML Text XHTML

The maximum operator



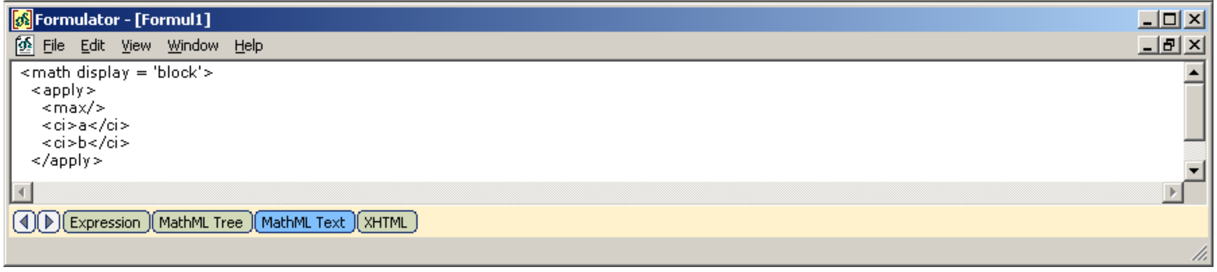
$\max\{\ \square\ ,\ \square\ \}$

Navigation: line - content MathML; parent node - "apply" ("max") element
Size: Regular Style: Math Zoom: 200 %



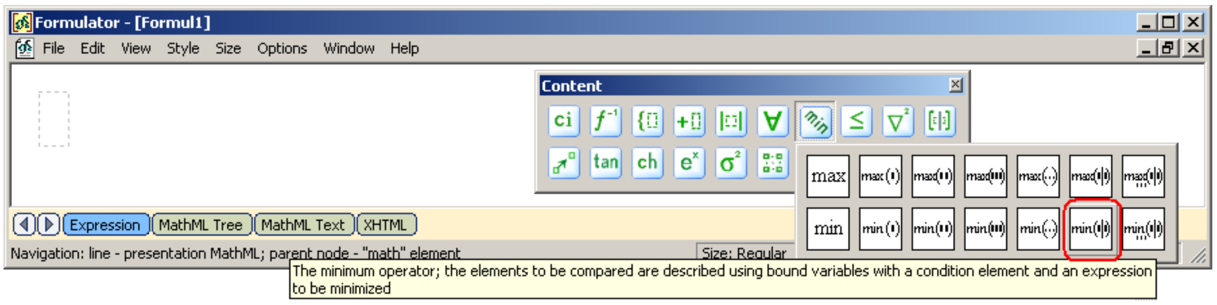
$\max\{\, a,\ b \,\}$

Navigation: line - content MathML; parent node - "apply" ("max") element
Size: Regular Style: Math Zoom: 200 %



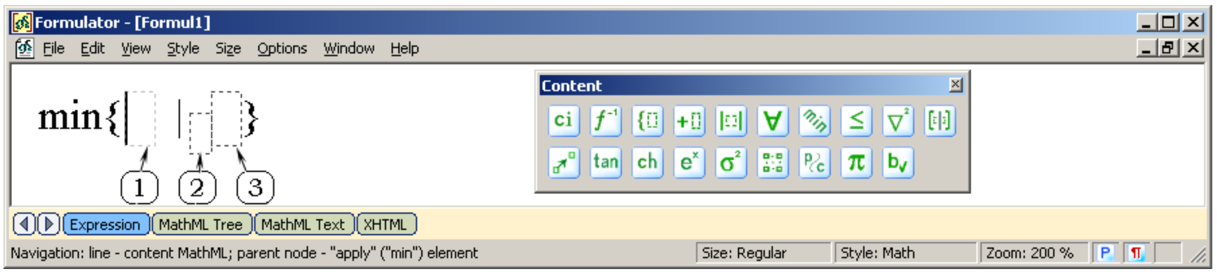
```
<math display = 'block'>
 <apply>
  <max/>
  <ci>a</ci>
  <ci>b</ci>
 </apply>
```

```
<apply>
  <min/>
  <bvar><ci>x</ci></bvar>
  <condition>
    <apply><notin/><ci> x </ci><ci type="set"> B </ci></apply>
  </condition>
  <apply>
      <power/>
    <ci> x </ci>
    <cn> 2 </cn>
  </apply>
</apply>
```

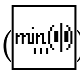


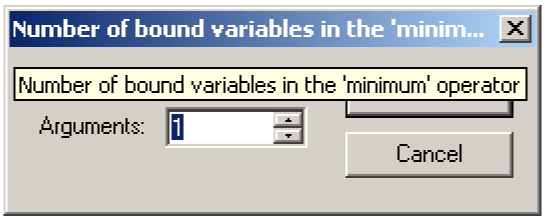
Navigation: line - presentation MathML; parent node - "math" element
Size: Regular

The minimum operator; the elements to be compared are described using bound variables with a condition element and an expression to be minimized



Navigation: line - content MathML; parent node - "apply" ("min") element
Size: Regular Style: Math Zoom: 200 %

Here (1) is an expression to be minimized; (2) is an input slot for one bound variable (the "bvar" element); (3) a condition element. There is also a button for many bounded variables () that shows the following dialog:


Number of bound variables in the 'minim...

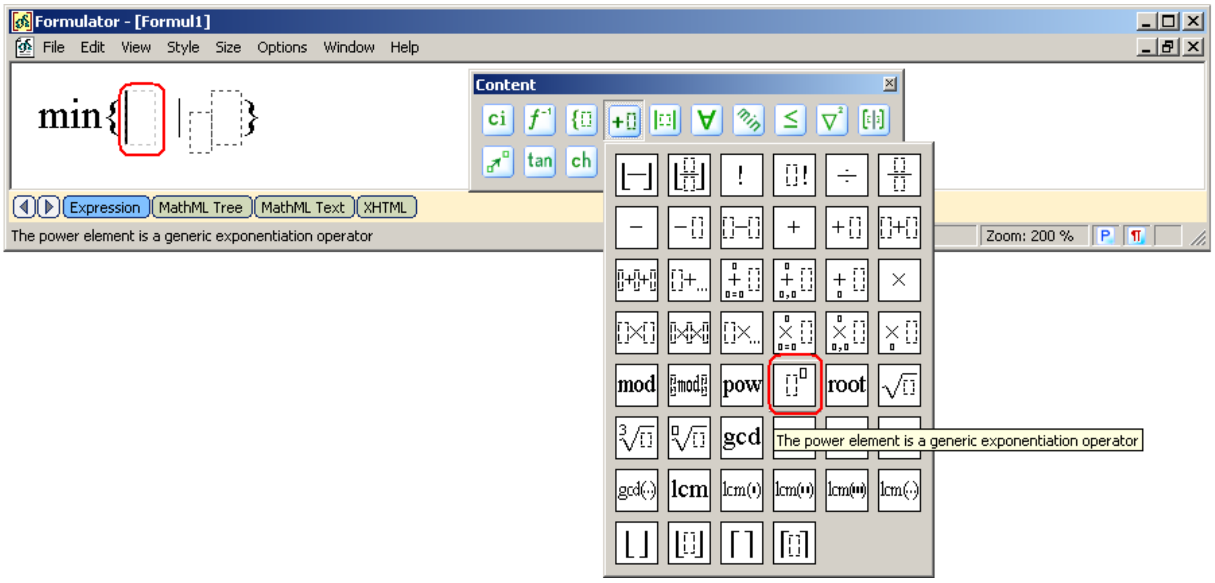
Number of bound variables in the 'minimum' operator

Arguments: 1

Cancel

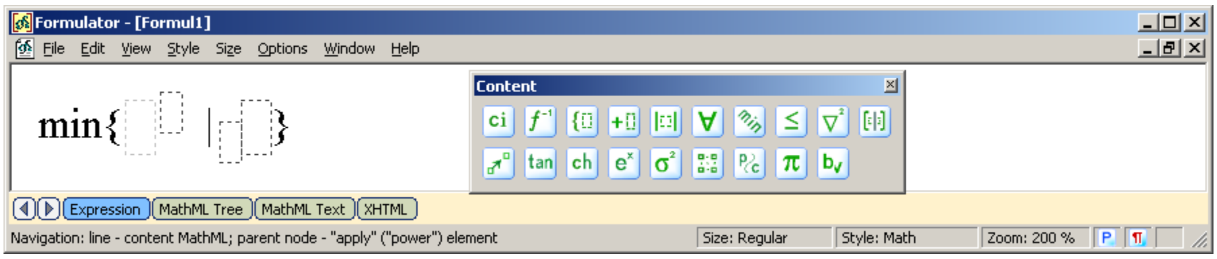
This approach is common for all the cases when qualifier elements of the MathML content markup are needed.

Please note that according to the W3C MathML Recommendation the "bvar" element should not be rendered in such a content markup element, so we consider it as an invisible (see the section "Basic Content Elements: invisible and transparent elements" for detailed discussion). This means that the corresponding input slot is available only after insertion of the "min" element, because currently rendering of invisible elements is turned off ( ).



The power element is a generic exponentiation operator

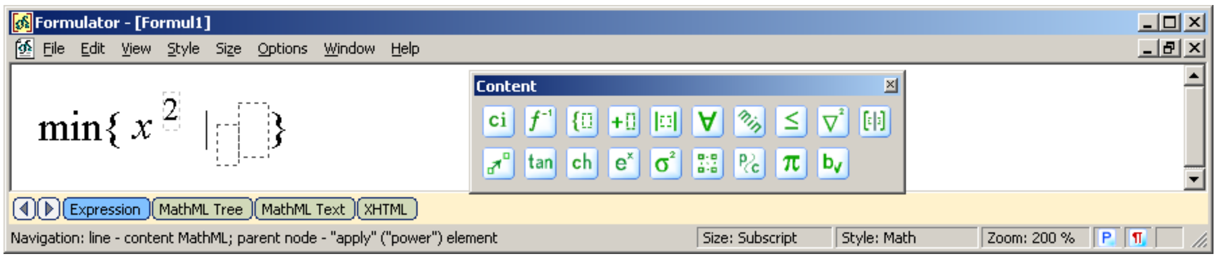
Insert the "apply" element with the <power/> operation.



Navigation: line - content MathML; parent node - "apply" ("power") element

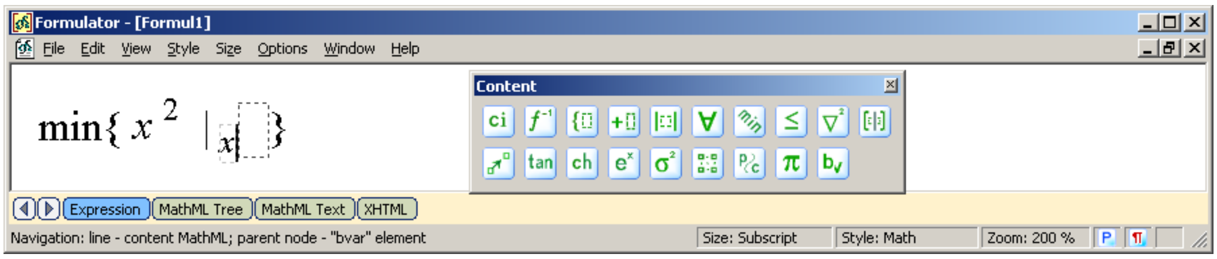
Size: Regular | Style: Math | Zoom: 200 %

Type "x", "2".


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Subscript Style: Math Zoom: 200 %

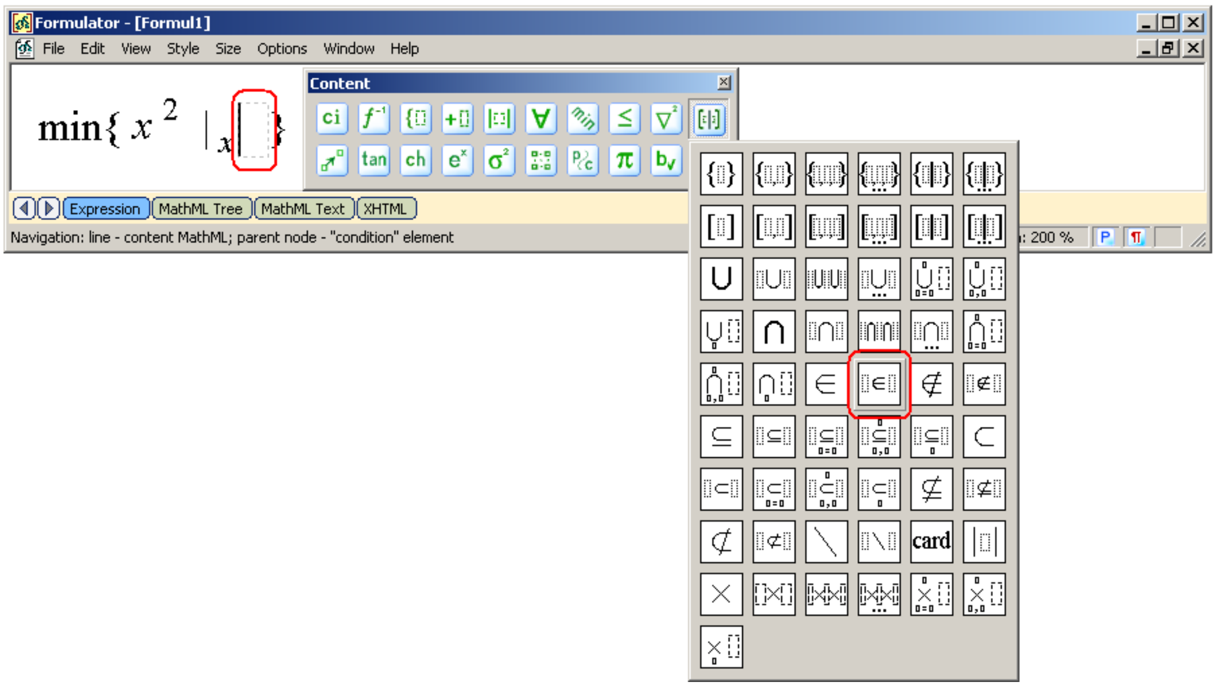
Navigation: line - content MathML; parent node - "apply" ("power") element

Navigate to the “bar” input slot and type a name of the bounded variable (“x”).

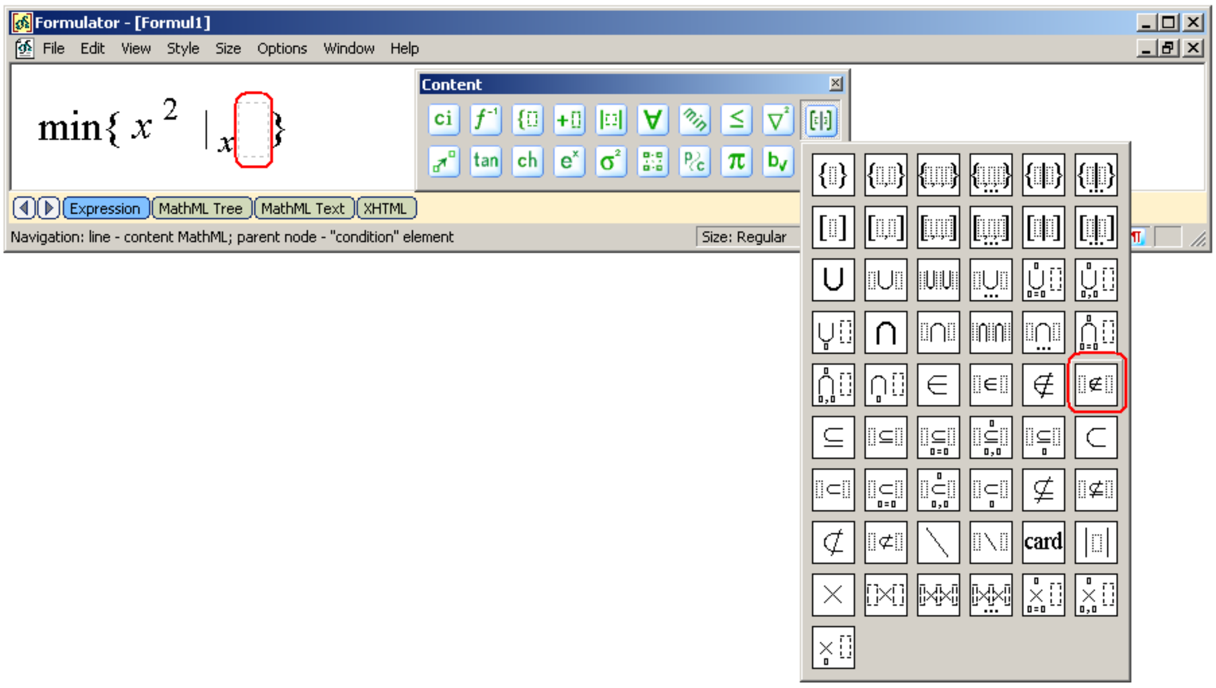

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Subscript Style: Math Zoom: 200 %

Navigation: line - content MathML; parent node - "bvar" element

Create an expression for the “condition” element.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
card

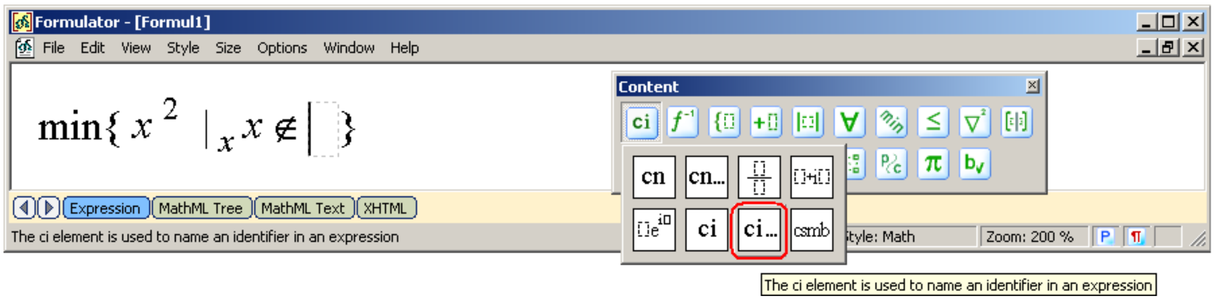
Navigation: line - content MathML; parent node - "condition" element


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
card

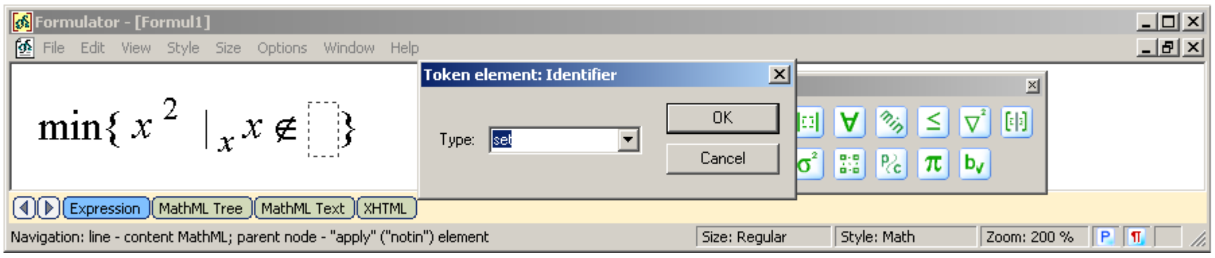
Navigation: line - content MathML; parent node - "condition" element

Size: Regular

Type "x" as a value of the first argument of the "apply" element in the condition; create the "ci" element of the "set" type as the second argument of the "apply".


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
cn
cn...
ci
ci...
csmb
Expression MathML Tree MathML Text XHTML
Style: Math
Zoom: 200 %

The ci element is used to name an identifier in an expression

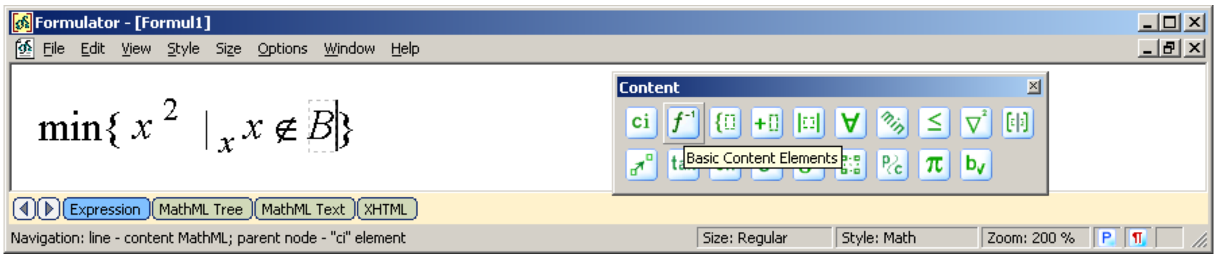

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Token element: Identifier
OK
Cancel
Expression MathML Tree MathML Text XHTML

Type: set

Navigation: line - content MathML; parent node - "apply" ("notin") element

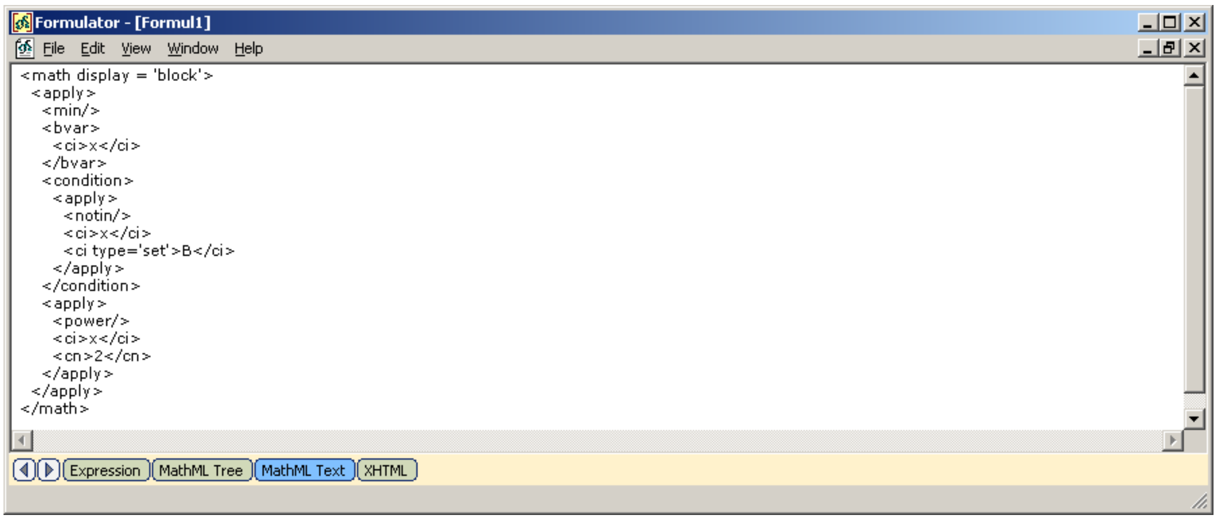
Size: Regular Style: Math Zoom: 200 %

See results on different pages of Formulator 3.7 MathML Weaver.

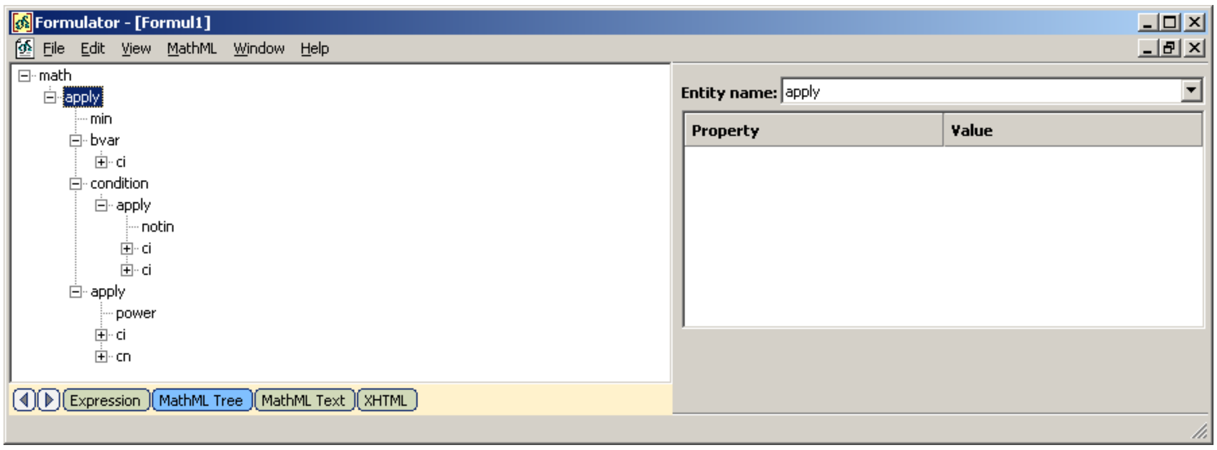

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Basic Content Elements
Expression MathML Tree MathML Text XHTML

Navigation: line - content MathML; parent node - "ci" element

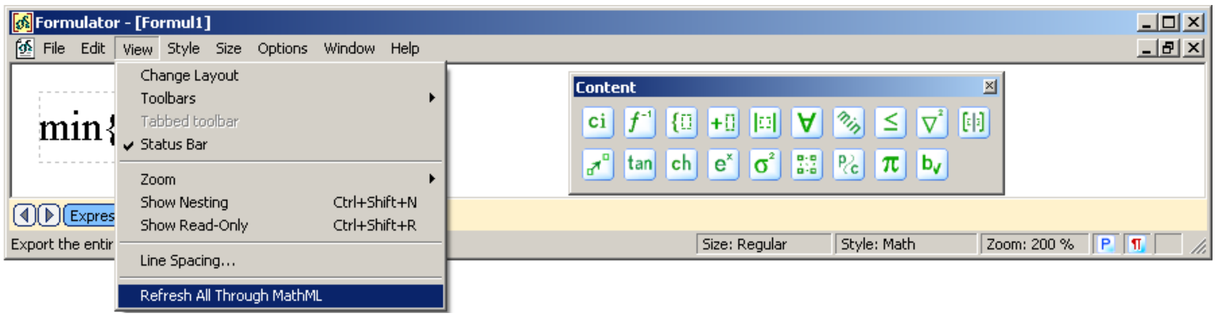
Size: Regular Style: Math Zoom: 200 %

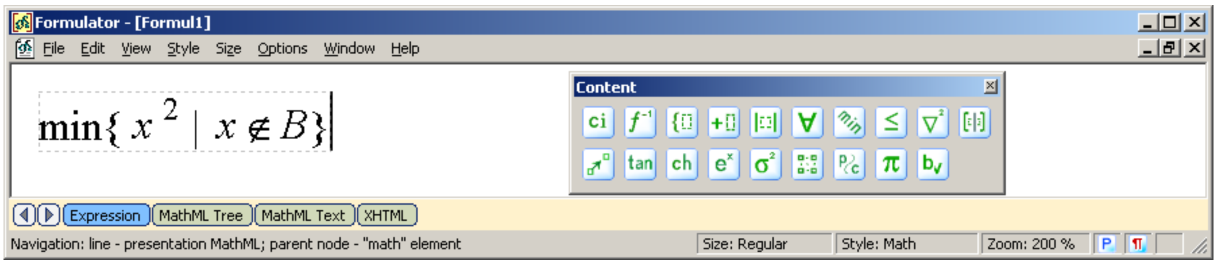


```
<math display = 'block'>
 <apply>
  <min/>
  <bvar>
   <ci>x</ci>
  </bvar>
  <condition>
   <apply>
    <notin/>
    <ci>x</ci>
    <ci type='set'>B</ci>
   </apply>
  </condition>
  <apply>
   <power/>
   <ci>x</ci>
   <cn>2</cn>
  </apply>
 </apply>
</math>
```



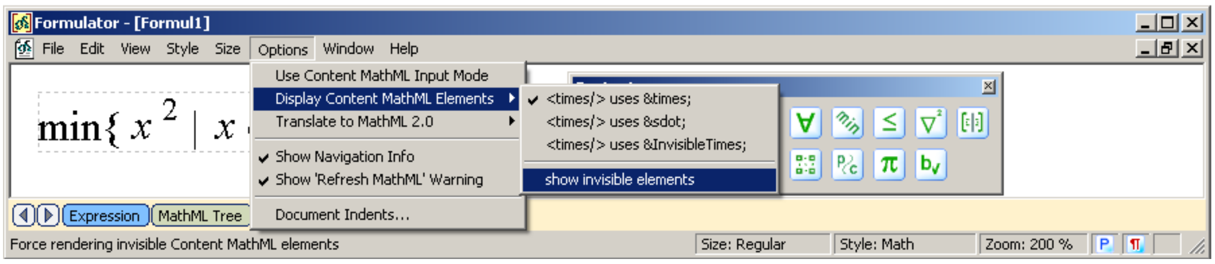
Now get back to the “Expression” page and refresh the document contents through MathML.



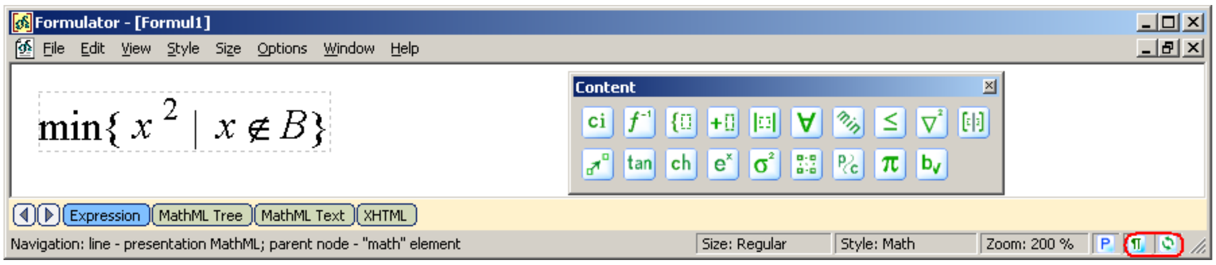

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression
MathML Tree
MathML Text
XHTML
Navigation: line - presentation MathML; parent node - "math" element
Size: Regular
Style: Math
Zoom: 200 %

$$\min\{\, x^{2} \mid x \notin B \}$$

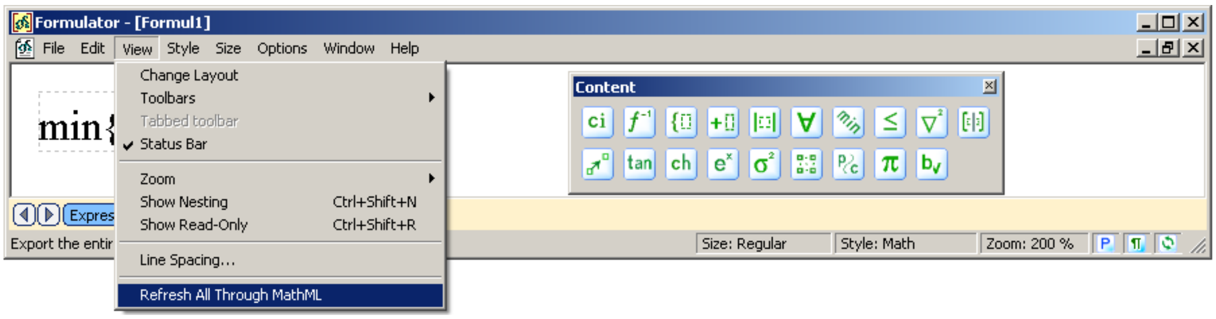
Note that the bounded variable is hided according to the proper rendering of the “min” element with condition qualifier. We can edit the bounded variable at any time if we turn on rendering of invisible elements and refresh the document through MathML once more.



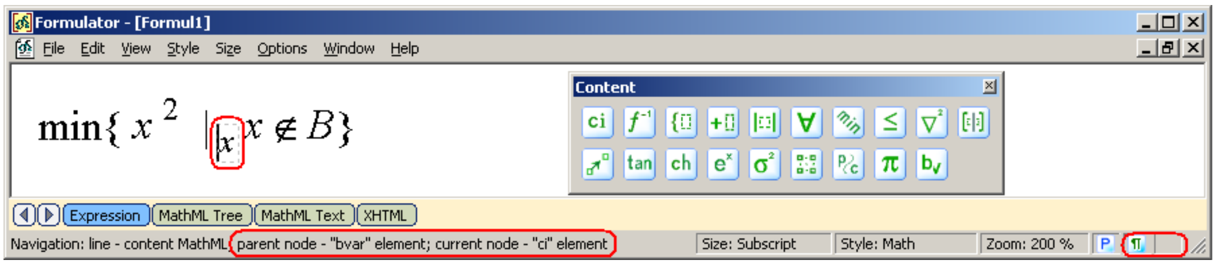
Force rendering invisible Content MathML elements



Navigation: line - presentation MathML; parent node - "math" element



Export the entir



Navigation: line - content MathML; parent node - "bvar" element; current node - "ci" element

2. Example 4.4.3.17.2 from the W3C MathML Recommendation): universal quantifier (forall):

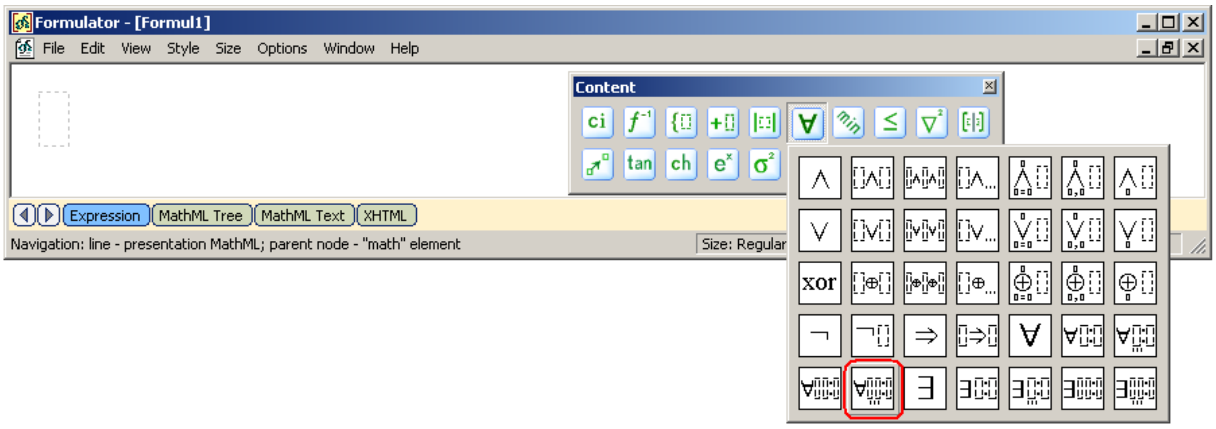
$$\forall\ p,\ q,\ p \in \mathbb{Q} \wedge q \in \mathbb{Q} \wedge p < q : p < q^{2}$$

```
<apply>
  <forall/>
  <bvar><ci> p </ci></bvar>
  <bvar><ci> q </ci></bvar>
  <condition>
    <apply><and/>
      <apply><in/><ci> p </ci><rationals/></apply>
      <apply><in/><ci> q </ci><rationals/></apply>
      <apply><lt/><ci> p </ci><ci> q </ci></apply>
    </apply>
  </condition>
  <apply><lt/>
      <ci> p </ci>
      <apply>
```

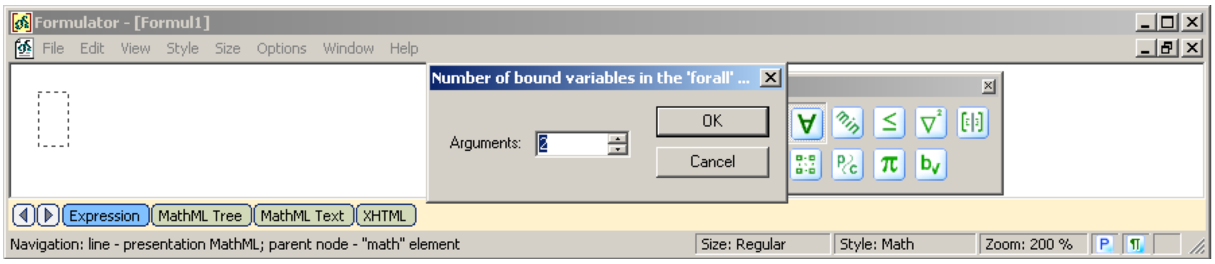
```
          <power/>
        <ci> q </ci>
        <cn> 2 </cn>
      </apply>
  </apply>
</apply>
```

The forall element is usually used in conjunction with one or more bound variables, an optional condition element, and an assertion. There is lot of available combination of different constructs of this element (this is true also for other similar statements, like the "exists" element, sums and products, integrals and so on). So MathML Weaver propose several buttons on mathematical toolbars, trying to make it faster creating common MathML trees.

In the case of our example there is the button for the "forall" element with several bounded variables and a conditional element: .



Navigation: line - presentation MathML; parent node - "math" element

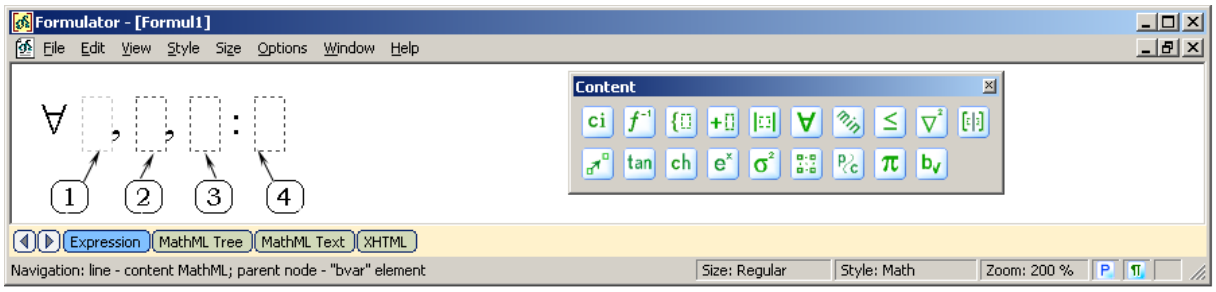


Number of bound variables in the 'forall' ...

Arguments: 2

Navigation: line - presentation MathML; parent node - "math" element

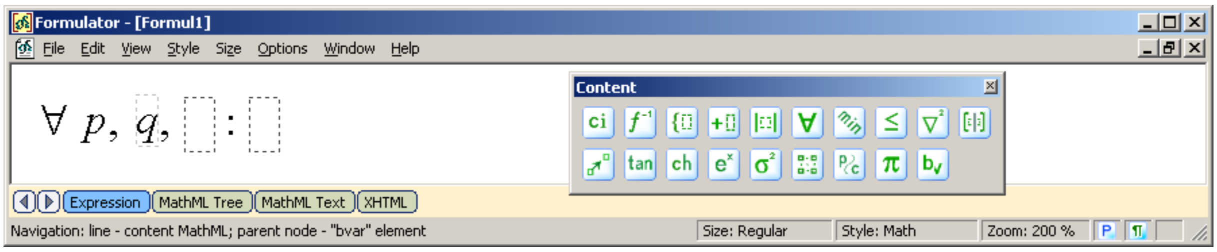
Size: Regular | Style: Math | Zoom: 200 %



Navigation: line - content MathML; parent node - "bvar" element

Size: Regular | Style: Math | Zoom: 200 %

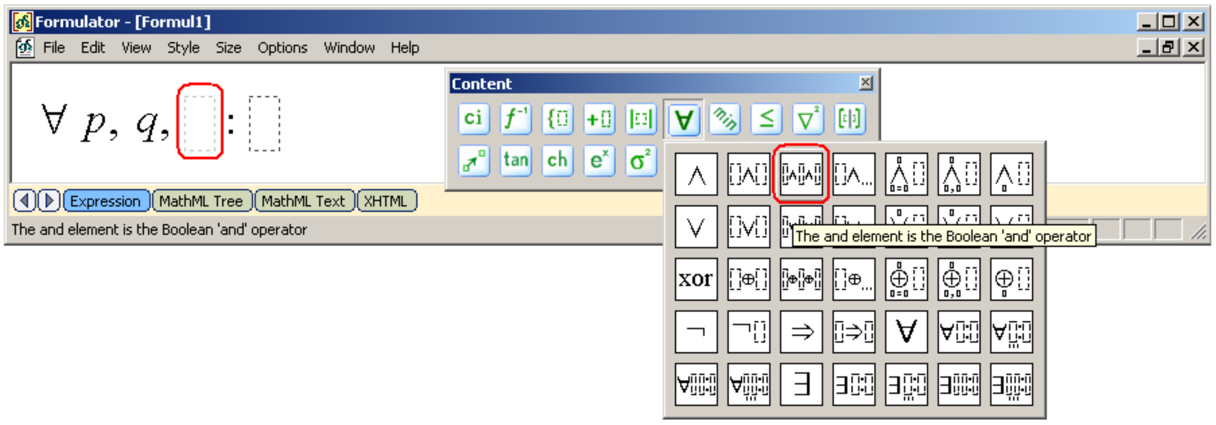
Here (1) and (2) are input slots for two bounded variables; (3) is a condition element; (4) is an assertion.


Formulator - [Formul1]

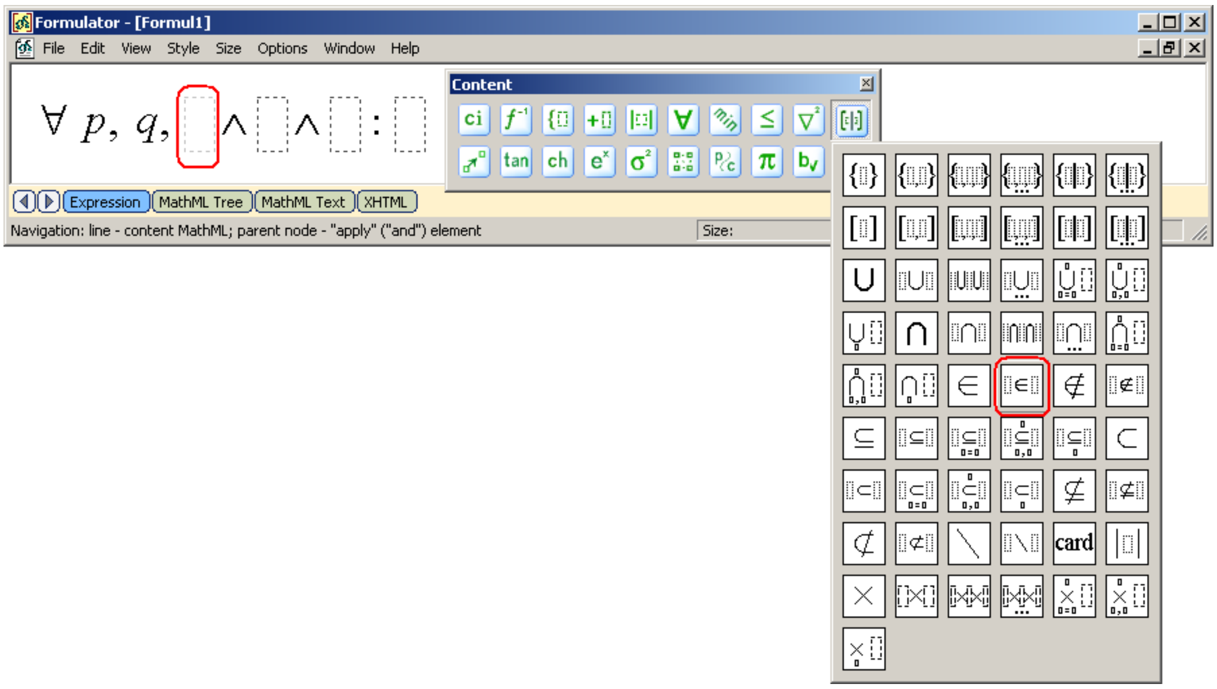
Navigation: line - content MathML; parent node - "bvar" element

Size: Regular Style: Math Zoom: 200 %

Now we should insert the "apply" element with the <and/> operation and three arguments.

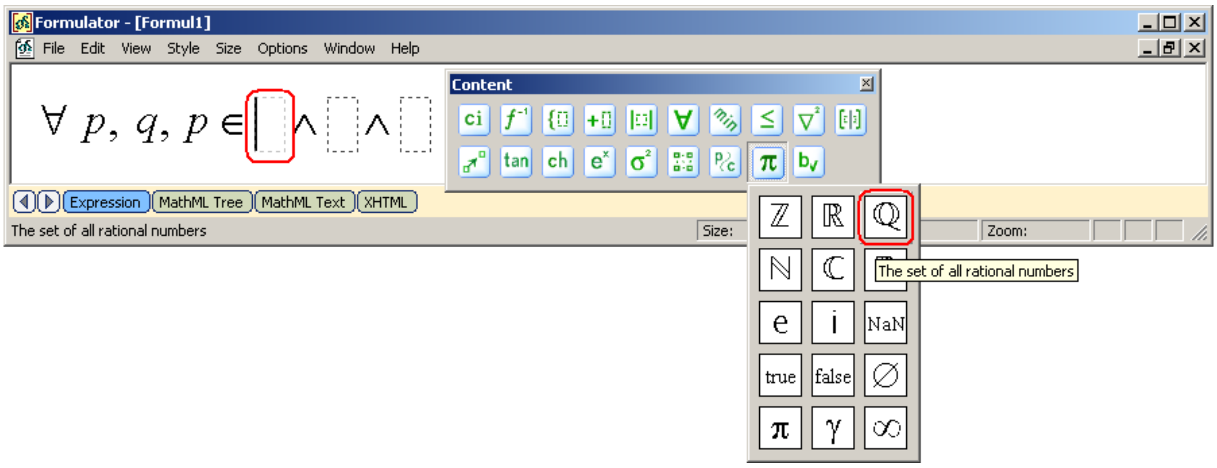

Formulator - [Formul1]

The and element is the Boolean 'and' operator


Formulator - [Formul1]

Navigation: line - content MathML; parent node - "apply" ("and") element

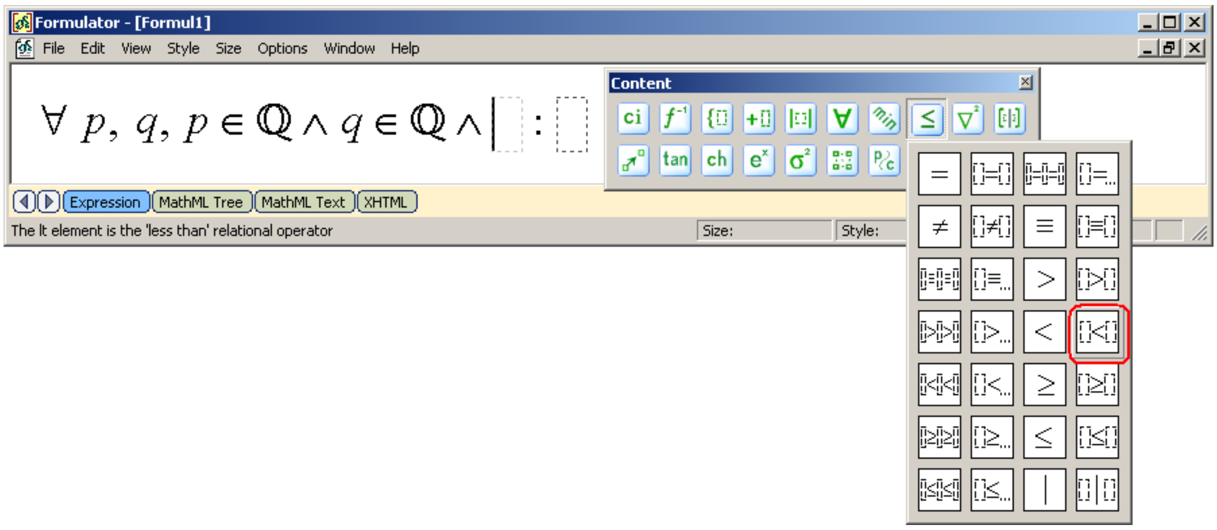
In order to insert the <rationals/> element use the π button:



Formulator - [Formul1]

$\forall\ p, q, p \in \square \wedge \square \wedge \square$

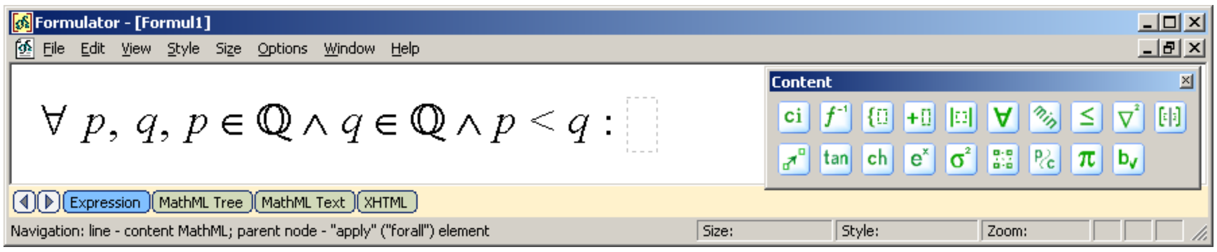
The set of all rational numbers



Formulator - [Formul1]

$\forall\ p, q, p \in \mathbb{Q} \wedge q \in \mathbb{Q} \wedge \square : \square$

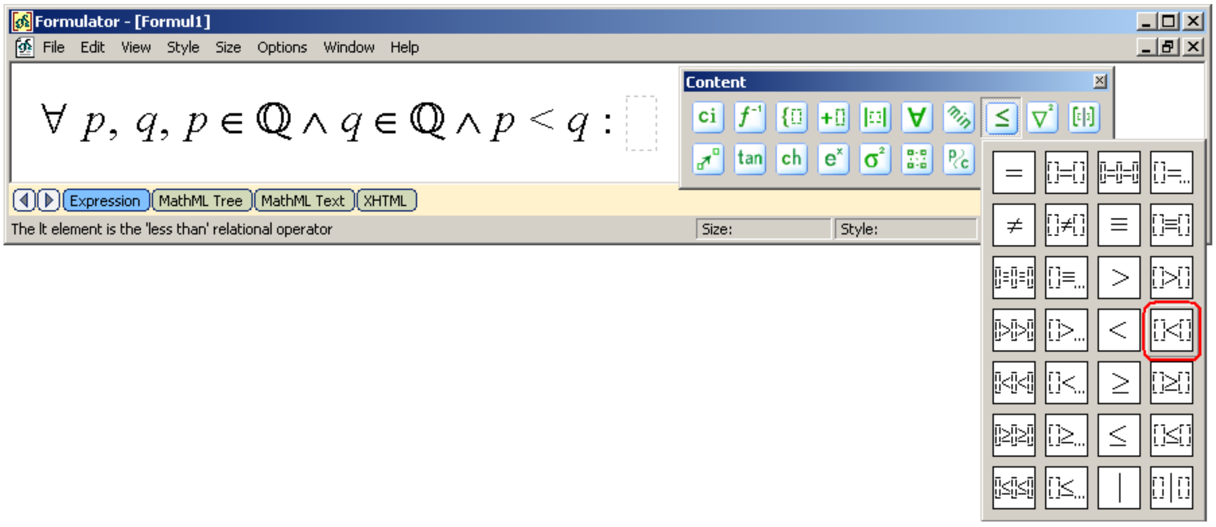
The lt element is the 'less than' relational operator



Formulator - [Formul1]

$\forall\ p, q, p \in \mathbb{Q} \wedge q \in \mathbb{Q} \wedge p < q : \square$

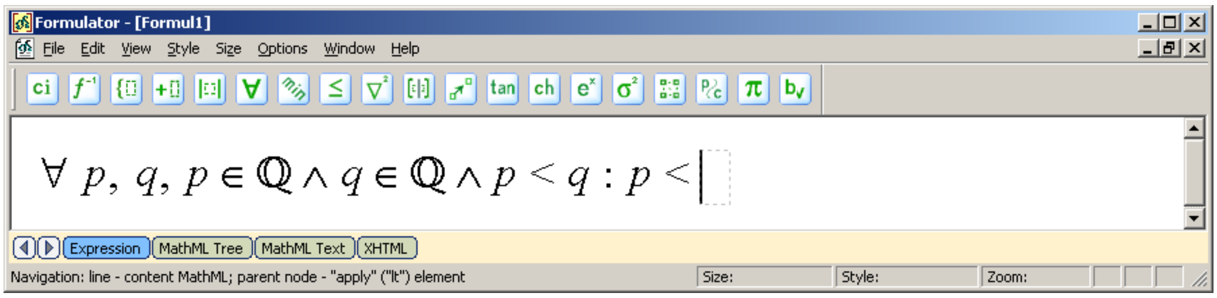
Navigation: line - content MathML; parent node - "apply" ("forall") element



Formulator - [Formul1]

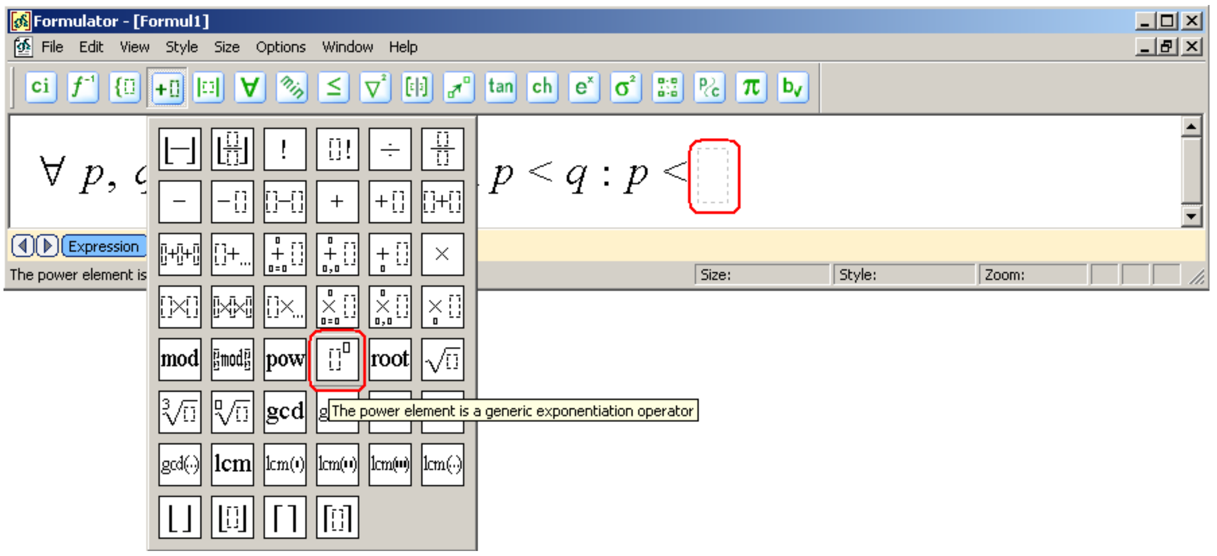
$\forall\ p, q, p \in \mathbb{Q} \wedge q \in \mathbb{Q} \wedge p < q : \square$

The lt element is the 'less than' relational operator


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Expression MathML Tree MathML Text XHTML

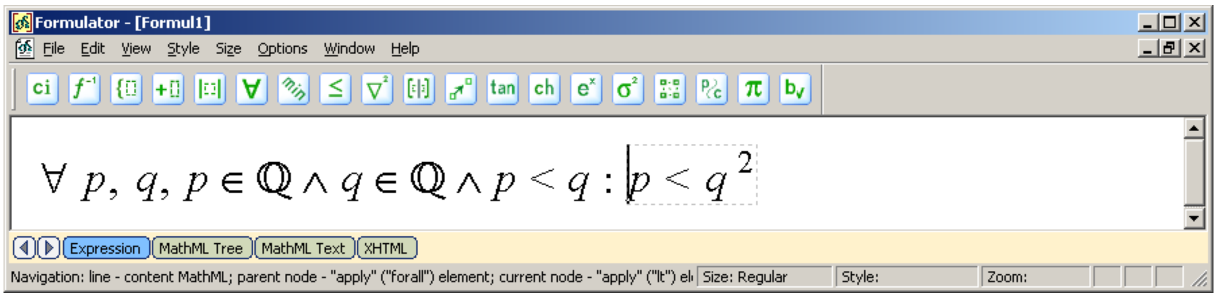
$\forall\, p, q, p \in \mathbb{Q} \wedge q \in \mathbb{Q} \wedge p < q : p <$

Navigation: line - content MathML; parent node - "apply" ("lt") element



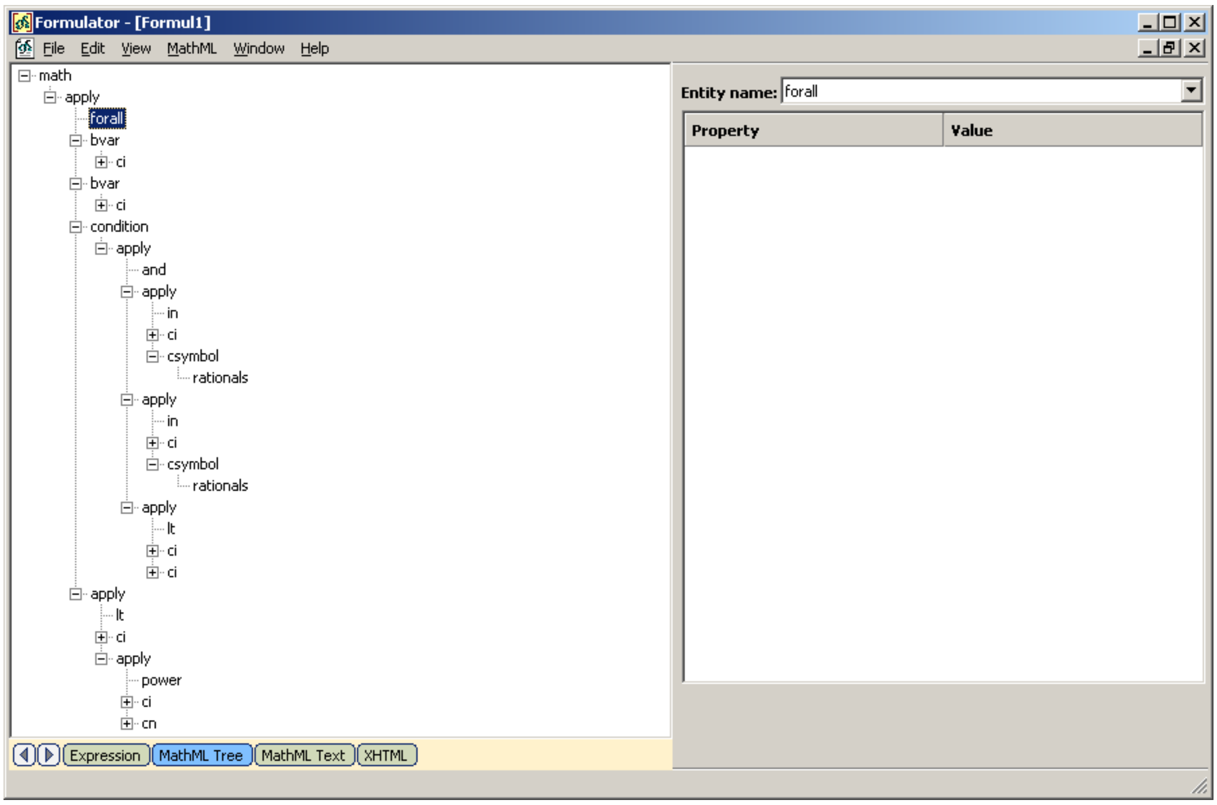
The power element is a generic exponentiation operator

See results on different pages of MathML Weaver.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Expression MathML Tree MathML Text XHTML

$\forall\, p, q, p \in \mathbb{Q} \wedge q \in \mathbb{Q} \wedge p < q : p < q^2$

Navigation: line - content MathML; parent node - "apply" ("forall") element; current node - "apply" ("lt") el Size: Regular

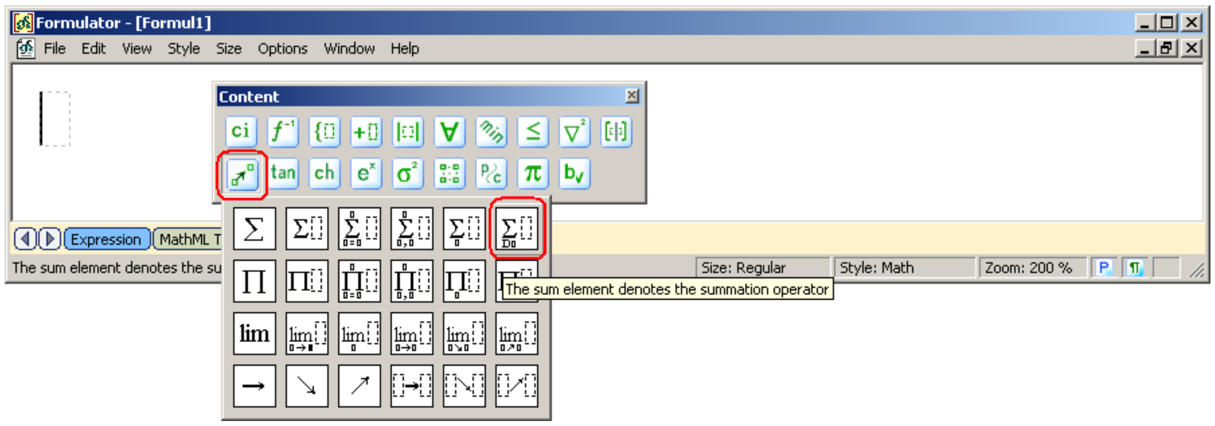

Formulator - [Formul1]
File Edit View MathML Window Help
Entity name: forall
Property
Value
Expression MathML Tree MathML Text XHTML

```
math
  apply
    forall
    bvar
      ci
    bvar
      ci
    condition
      apply
        and
        apply
          in
          ci
          csymbol
            rationals
        apply
          in
          ci
          csymbol
            rationals
        apply
          lt
          ci
          ci
    apply
      lt
      ci
      apply
        power
        ci
        cn
```

Sequences and Series

1. Example 4.4.7.1.2 from the W3C MathML Recommendation): the sum element:

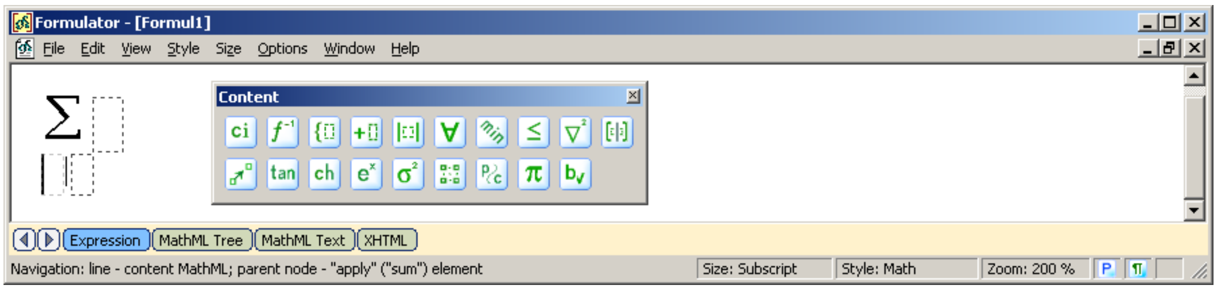
```
<apply>
  <sum/>
  <domainofapplication>
    <ci type="set"> B </ci>
  </domainofapplication>
  <ci type="function"> f </ci>
</apply>
```



The sum element denotes the su

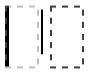
The sum element denotes the summation operator

Now it is inserted a visual form for editing the "sum" element with the "domainofapplication" qualifier.

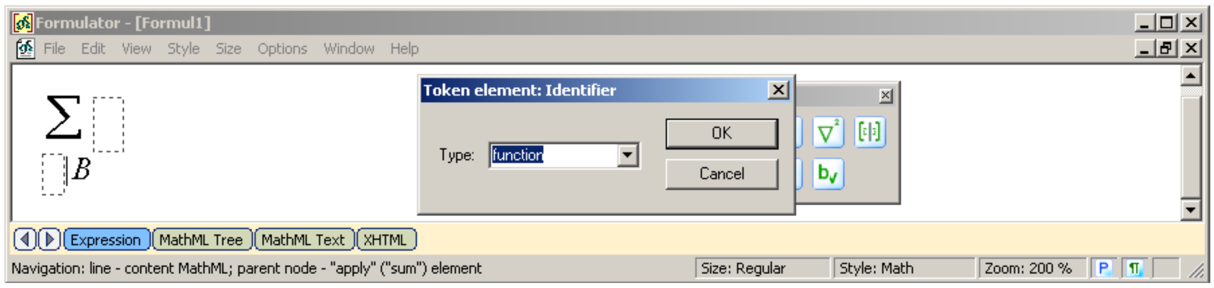

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci tan ch π
Expression MathML Tree MathML Text XHTML

Navigation: line - content MathML; parent node - "apply" ("sum") element | Size: Subscript | Style: Math | Zoom: 200 %

Input slots below the sum sign are rendered as

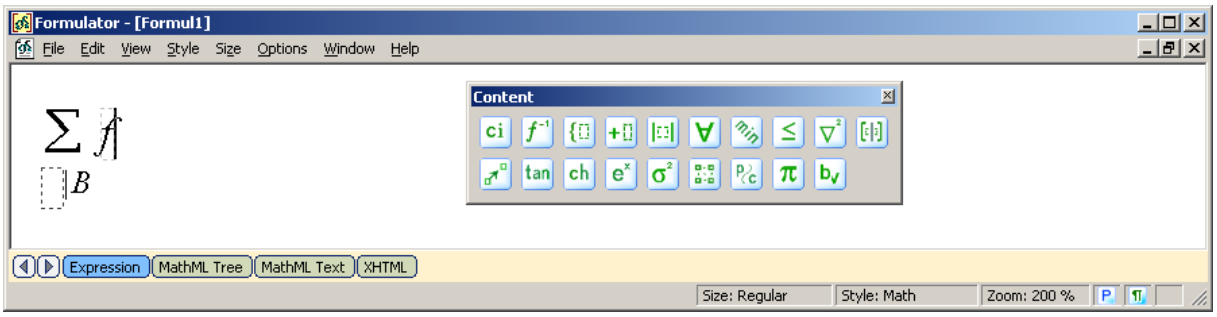


in order to allow to a user editing both the bounded variable (the first input slot) and the “domainofapplication” element. Leave the first input slot empty and create a “ci” element of type “set” for the second input slot. Then create another “ci” element for the argument of the “sum” element.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Token element: Identifier
OK
Cancel
Expression MathML Tree MathML Text XHTML

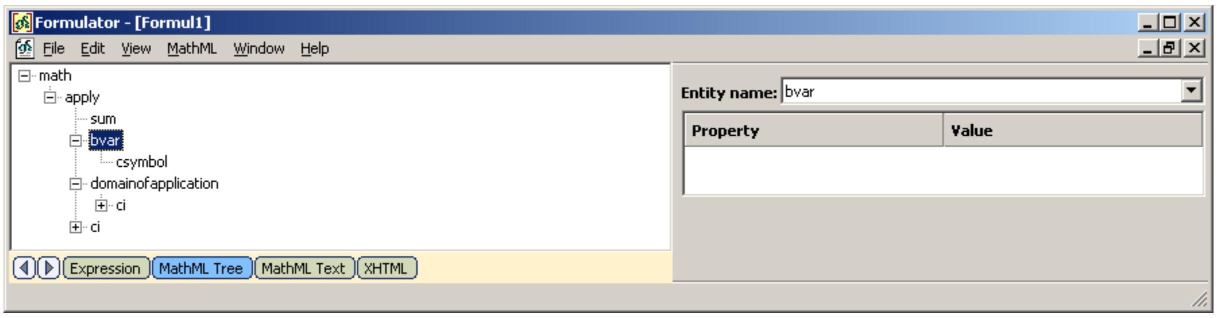
Type: function

Navigation: line - content MathML; parent node - "apply" ("sum") element | Size: Regular | Style: Math | Zoom: 200 %


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci tan ch π
Expression MathML Tree MathML Text XHTML

Size: Regular | Style: Math | Zoom: 200 %

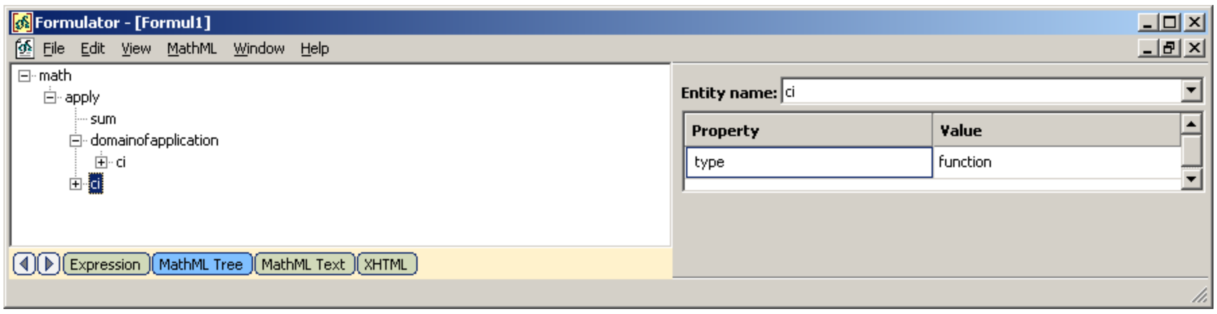
Switch to the “MathML Tree” page and delete the node for the bounded variable, since the example don’t need the “bvar” presence. In order to do this, select the “bvar” node and press Delete.


Formulator - [Formul1]
File Edit View MathML Window Help
Expression MathML Tree MathML Text XHTML

```
math
  apply
    sum
    bvar
      csymbol
    domainofapplication
      ci
    ci
```

Entity name: bvar

Property	Value

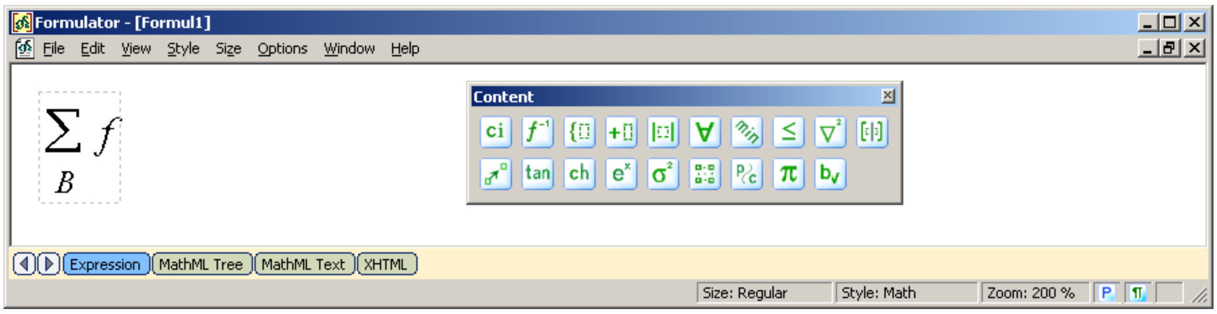

Formulator - [Formul1]
File Edit View MathML Window Help
Expression MathML Tree MathML Text XHTML

math
apply
sum
domainofapplication
ci
ci

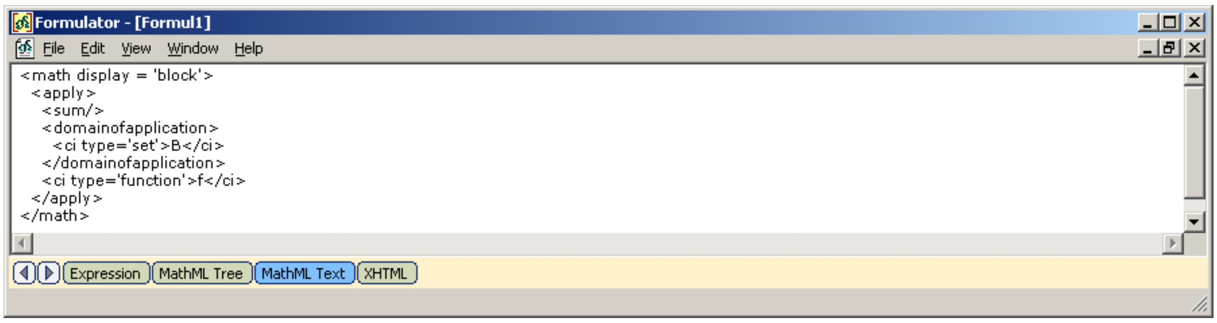
Entity name: ci

Property	Value
type	function

Then get to the “Expression” and “MathML Text” pages to see results.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

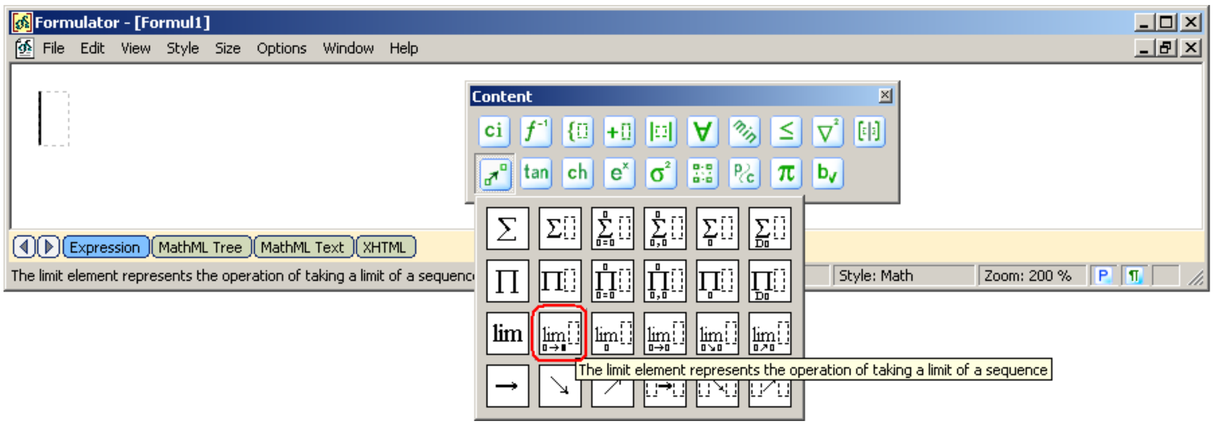
$$\sum_B f$$


Formulator - [Formul1]
File Edit View Window Help
Expression MathML Tree MathML Text XHTML

```
<math display = 'block'>
 <apply>
  <sum/>
  <domainofapplication>
   <ci type='set'>B</ci>
  </domainofapplication>
  <ci type='function'>f</ci>
 </apply>
</math>
```

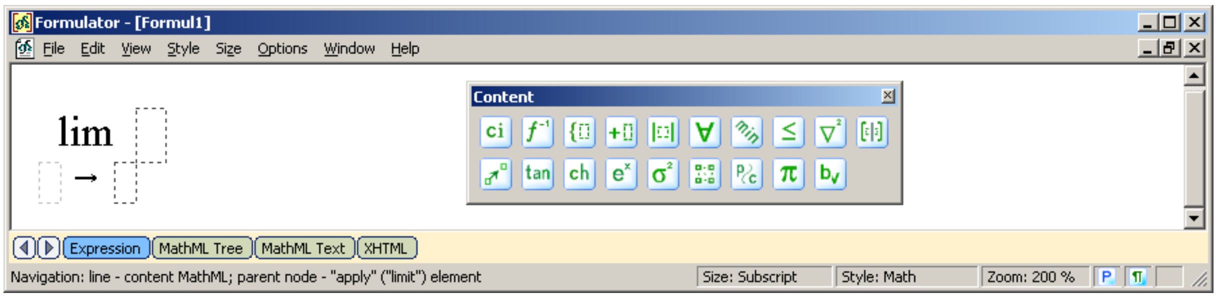
2. Examples 4.4.7.3.2 from the W3C MathML Recommendation): the limit element:

```
<apply>
  <limit/>
  <bvar><ci> x </ci></bvar>
  <lowlimit><cn> 0 </cn></lowlimit>
  <apply><sin/><ci> x </ci></apply>
</apply>
```


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Style: Math
Zoom: 200 %

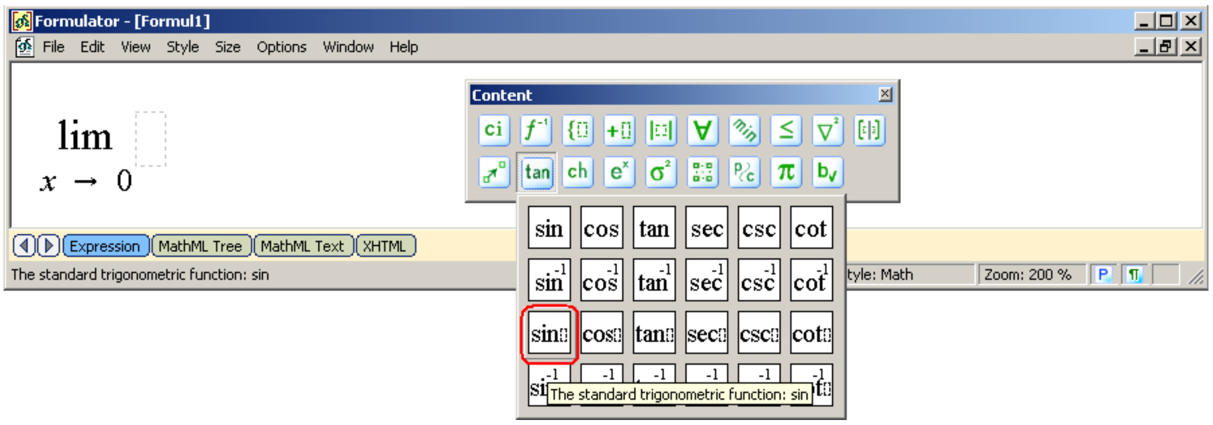
The limit element represents the operation of taking a limit of a sequenc

The limit element represents the operation of taking a limit of a sequence


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

Navigation: line - content MathML; parent node - "apply" ("limit") element

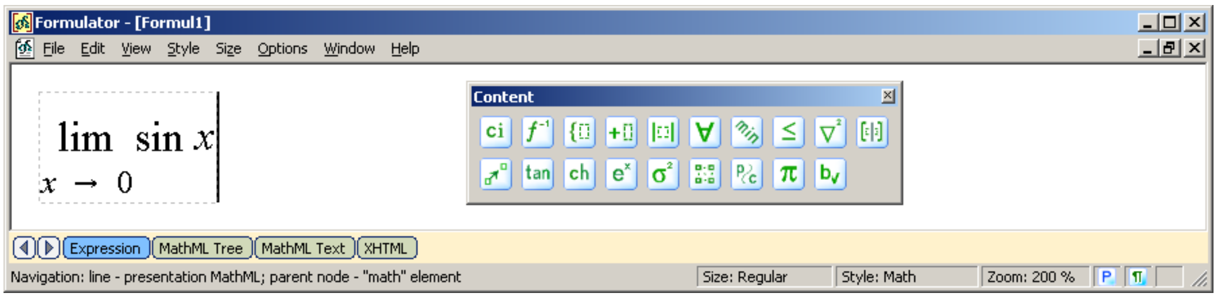
Size: Subscript Style: Math Zoom: 200 %


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
sin cos tan sec csc cot

The standard trigonometric function: sin

Style: Math Zoom: 200 %

The standard trigonometric function: sin

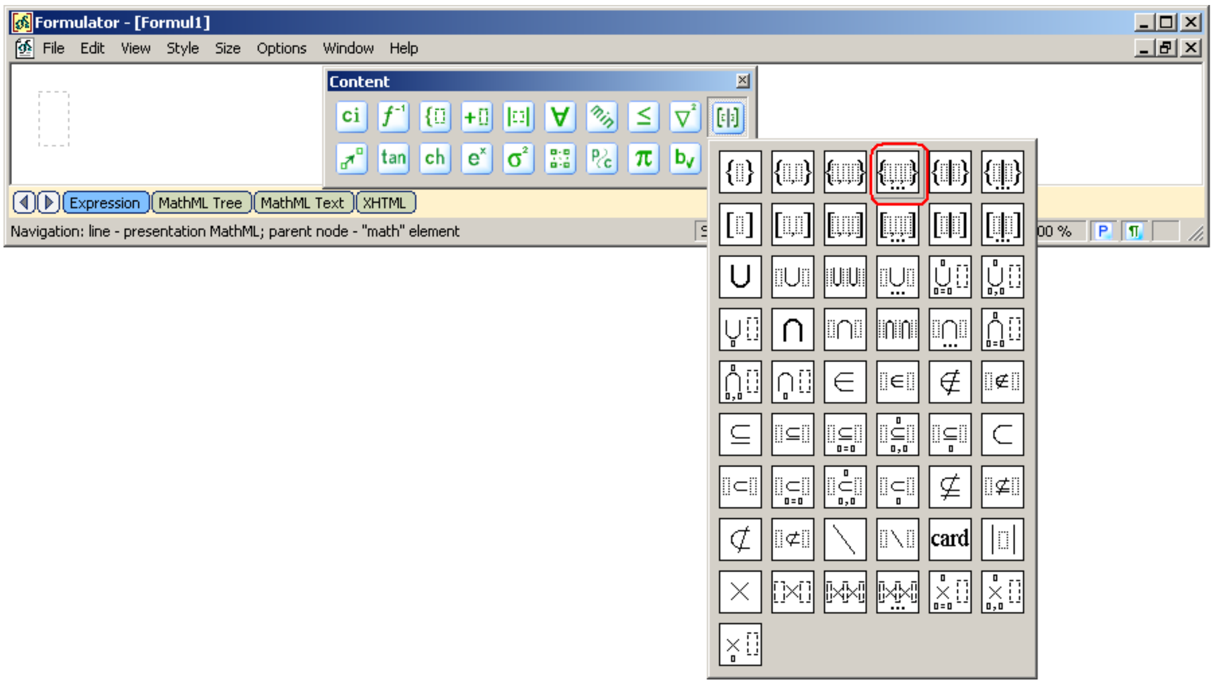

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

Navigation: line - presentation MathML; parent node - "math" element

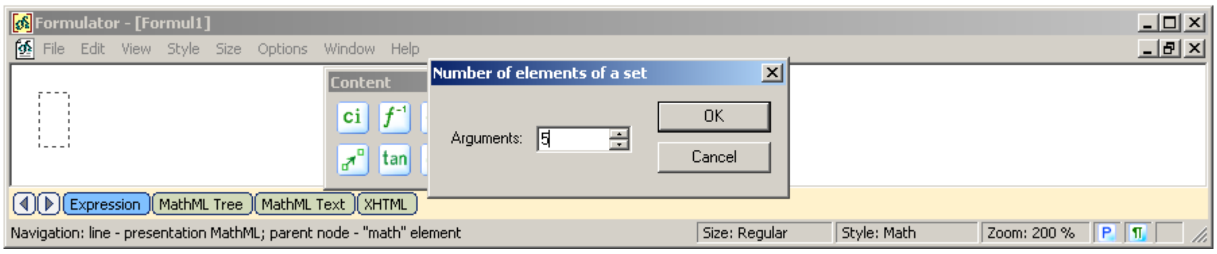
Size: Regular Style: Math Zoom: 200 %

Constructor elements: theory of sets and linear algebra

1. The next example shows how to create elements of the theory of sets.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
card

Navigation: line - presentation MathML; parent node - "math" element

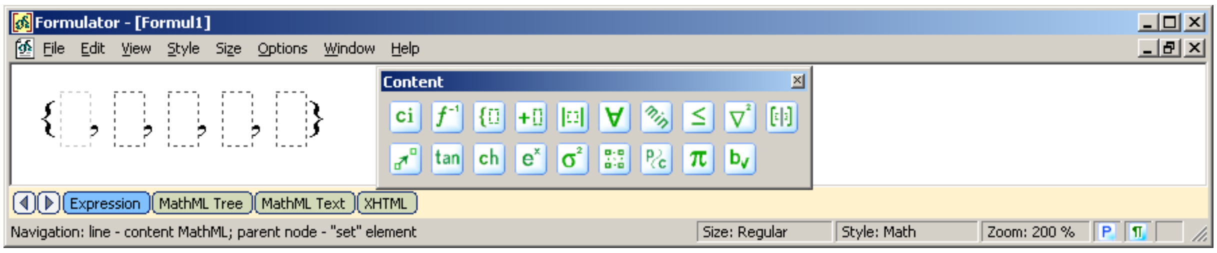

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
OK
Cancel
Expression MathML Tree MathML Text XHTML

Number of elements of a set

Arguments: 5

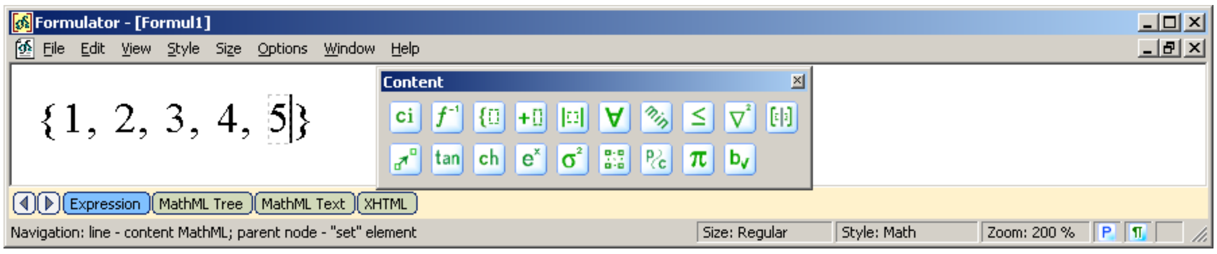
Navigation: line - presentation MathML; parent node - "math" element

Size: Regular Style: Math Zoom: 200 %


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

Navigation: line - content MathML; parent node - "set" element

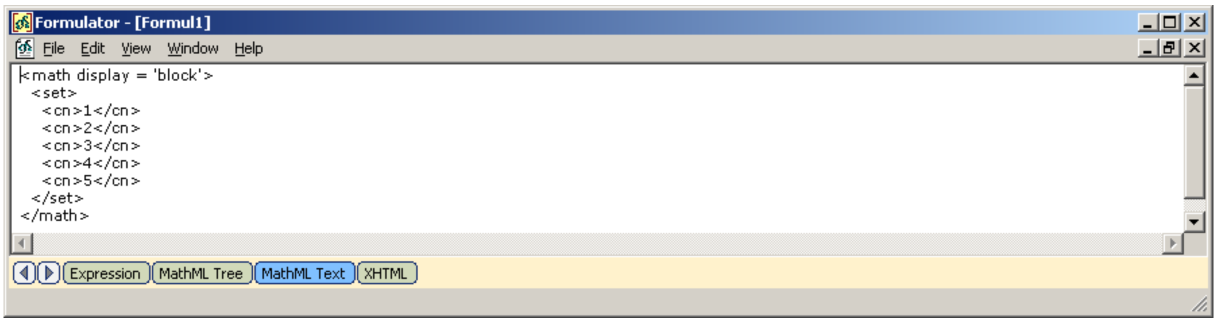
Size: Regular Style: Math Zoom: 200 %


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

$\{1,\ 2,\ 3,\ 4,\ 5\}$

Navigation: line - content MathML; parent node - "set" element

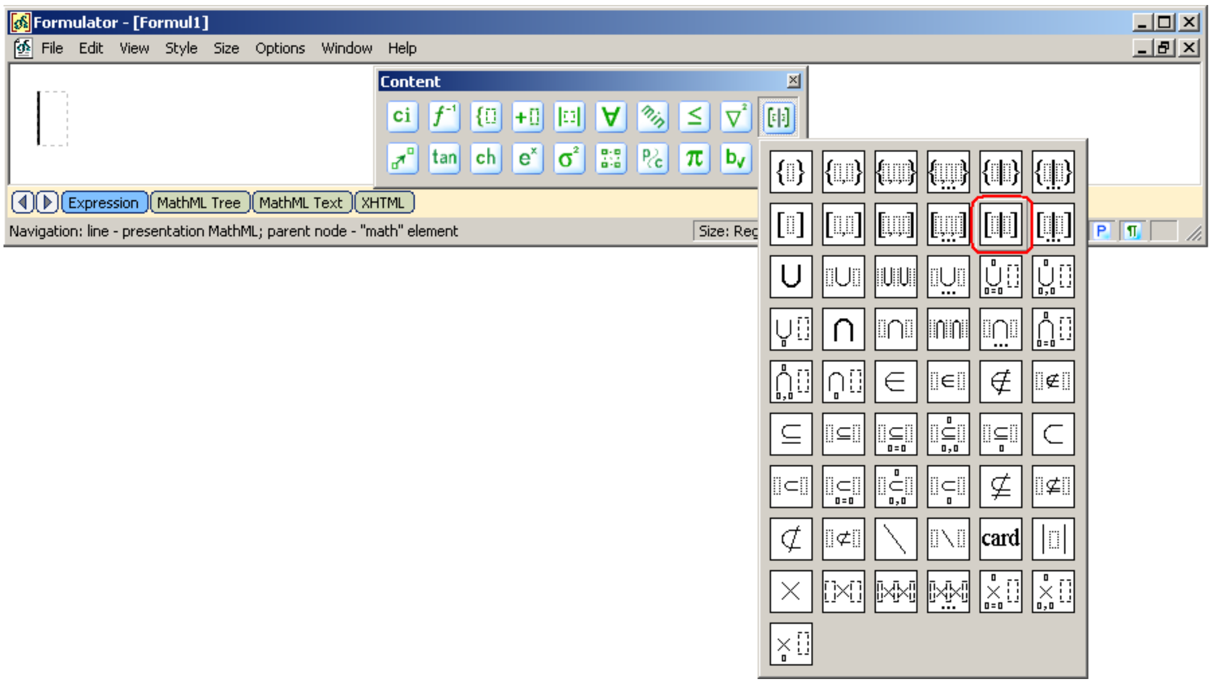
Size: Regular Style: Math Zoom: 200 %


Formulator - [Formul1]
File Edit View Window Help
Expression MathML Tree MathML Text XHTML

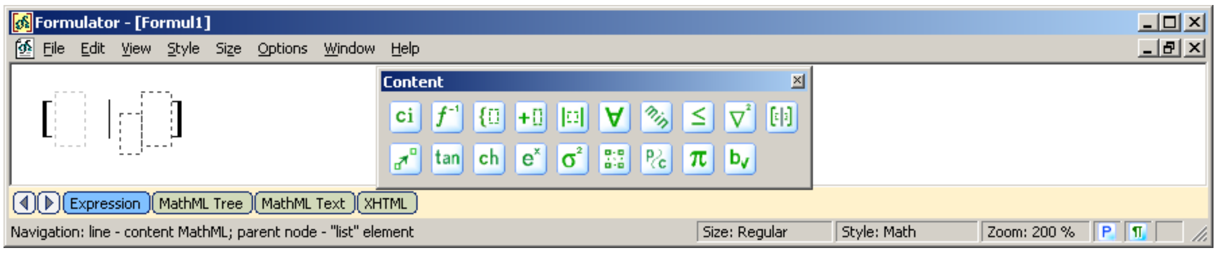
```
<math display = 'block'>
  <set>
    <cn>1</cn>
    <cn>2</cn>
    <cn>3</cn>
    <cn>4</cn>
    <cn>5</cn>
  </set>
</math>
```

Now create a list using a bounded variable and a condition element (example 4.4.6.2.2 from the W3C MathML Recommendation):

```
<list order="numeric">
  <bvar><ci> x </ci></bvar>
  <condition>
    <apply><lt/>
      <ci> x </ci>
      <cn> 5 </cn>
    </apply>
  </condition>
  <ci> x </ci>
</list>
```

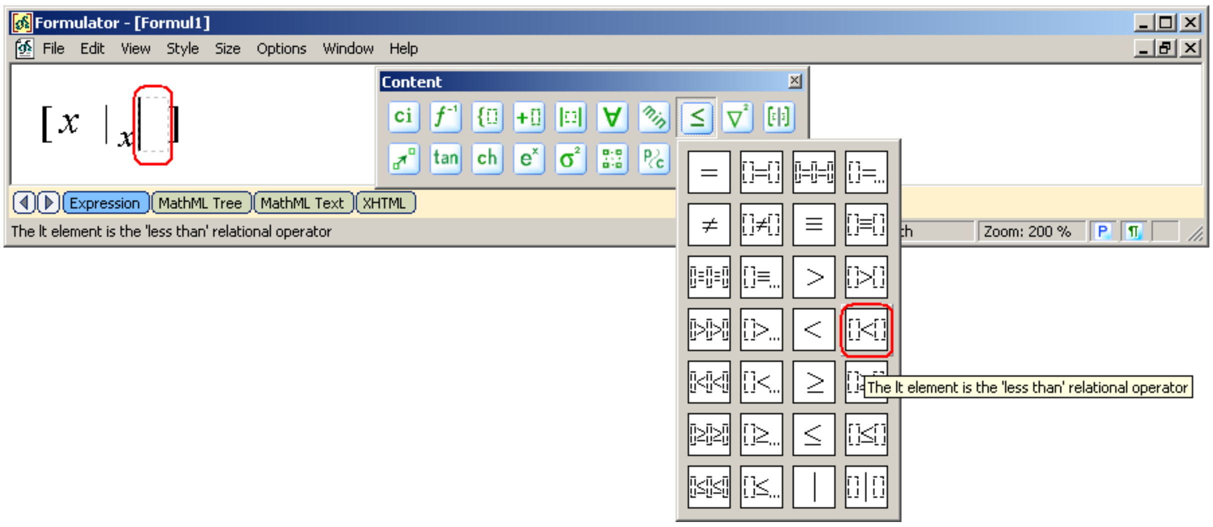

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Reg
card

Navigation: line - presentation MathML; parent node - "math" element

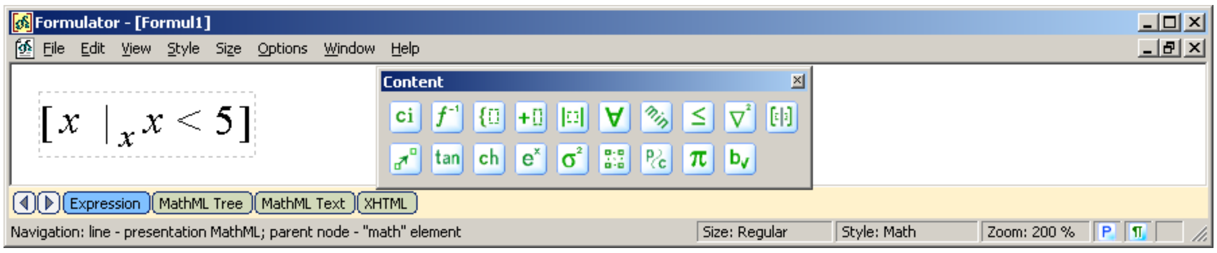

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

Navigation: line - content MathML; parent node - "list" element Size: Regular Style: Math Zoom: 200 %

If we recall the case of the minimum operator from the "Arithmetic, Algebra, Logic and Relations" section, it will be obvious that the structure of the just created "list" element is quite similar to that case. Namely, the starting input slot is for a rule of constructing list's items; after the vertical line there is a smaller input slot for a bounded variable; the last input slot is for a condition.

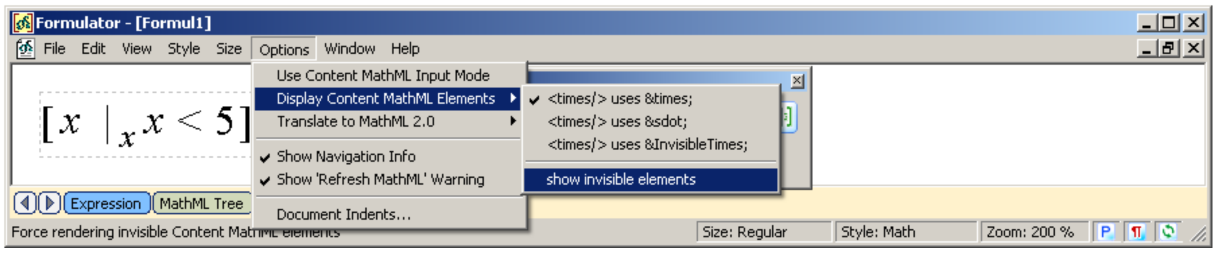

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci tan ch
Expression MathML Tree MathML Text XHTML
Zoom: 200 %

The lt element is the 'less than' relational operator

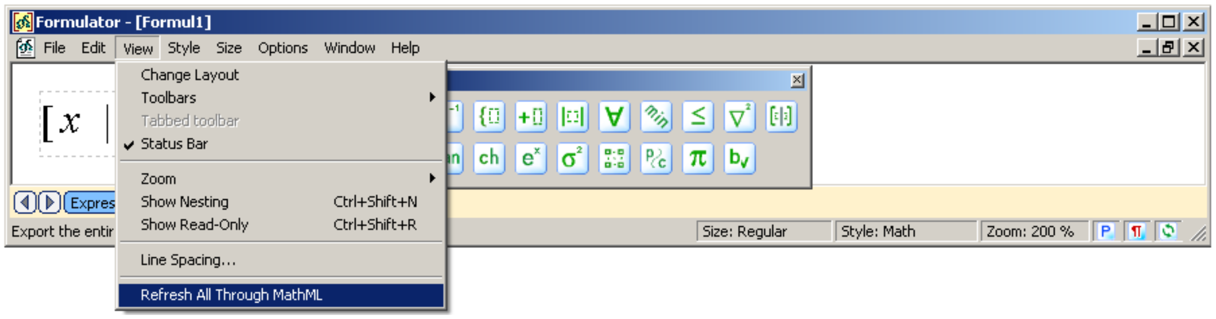

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

Navigation: line - presentation MathML; parent node - "math" element

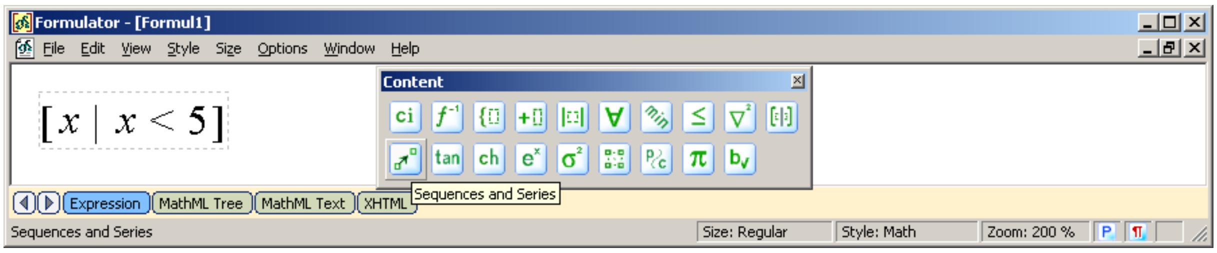
In order to get proper rendering of the list we should turn off the invisible elements rendering and refresh the document through MathML, as it is shown on the next figures.



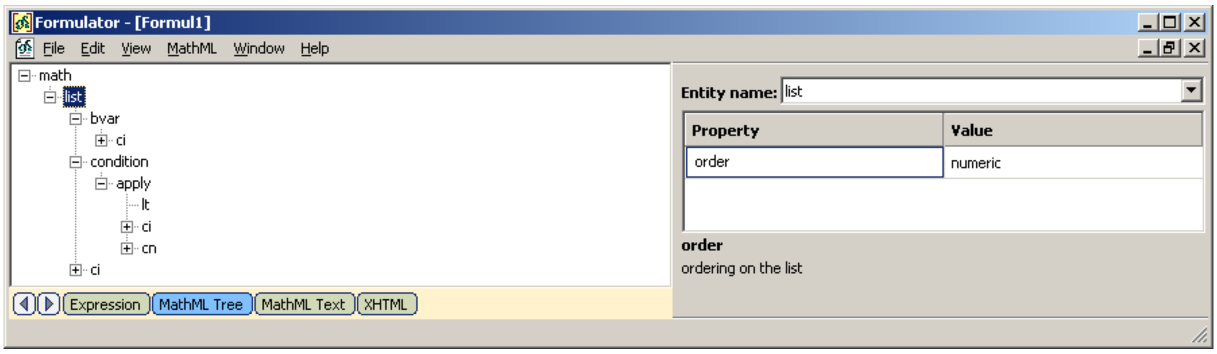
Force rendering invisible Content MathML elements


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Change Layout
Toolbars
Tabbed toolbar
Status Bar
Zoom
Show Nesting Ctrl+Shift+N
Show Read-Only Ctrl+Shift+R
Line Spacing…
Refresh All Through MathML
Size: Regular
Style: Math
Zoom: 200 %

Export the entir

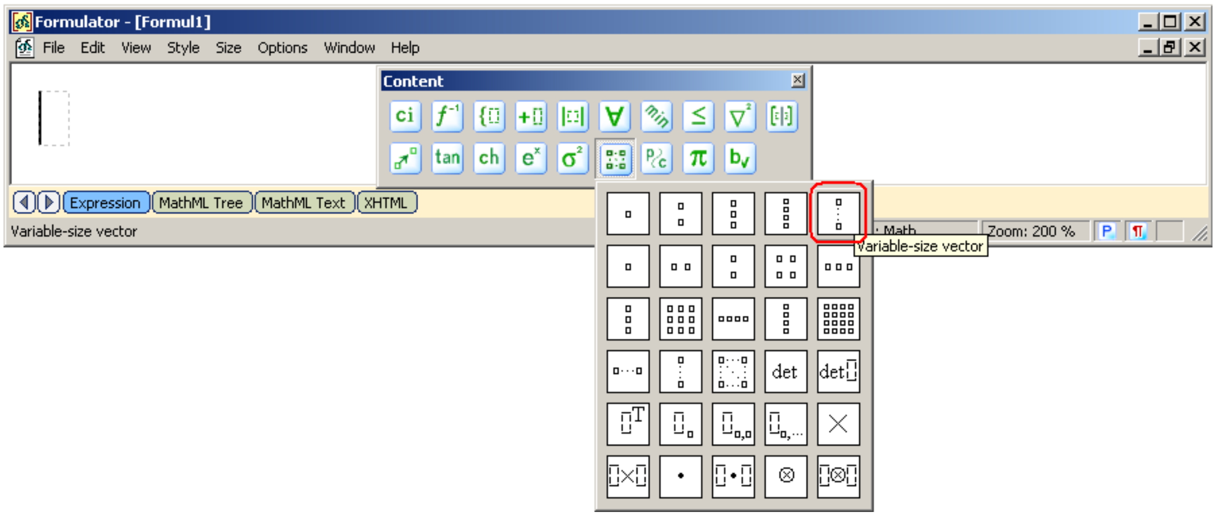

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
[x | x < 5]
Sequences and Series
Expression MathML Tree MathML Text XHTML
Sequences and Series
Size: Regular Style: Math Zoom: 200 %

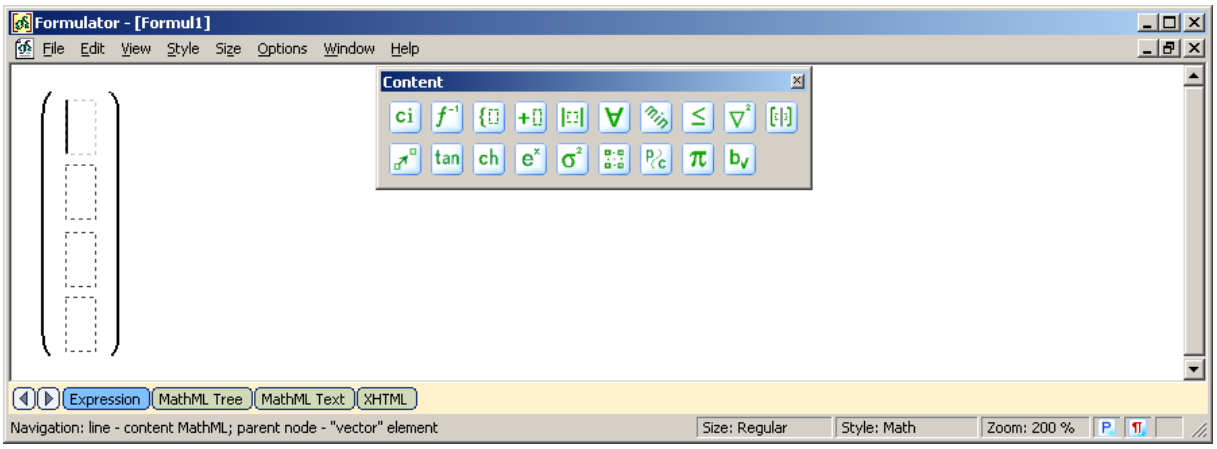
Switch to the "MathML Tree" page to see results.



2. The next examples shows how to create elements of the linear algebra.

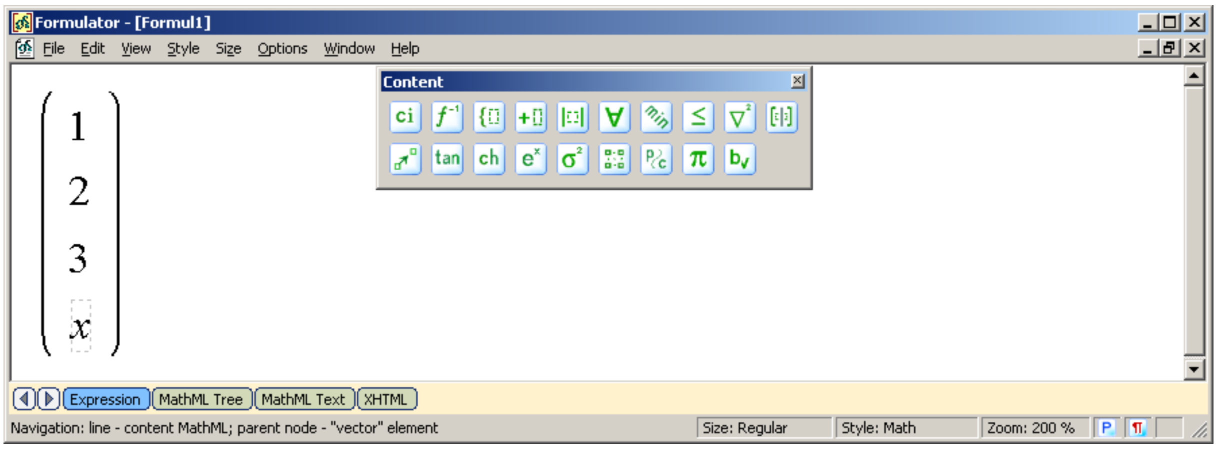
```
<vector>
  <cn> 1 </cn>
  <cn> 2 </cn>
  <cn> 3 </cn>
  <ci> x </ci>
</vector>
```




Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

Navigation: line - content MathML; parent node - "vector" element   Size: Regular   Style: Math   Zoom: 200 %

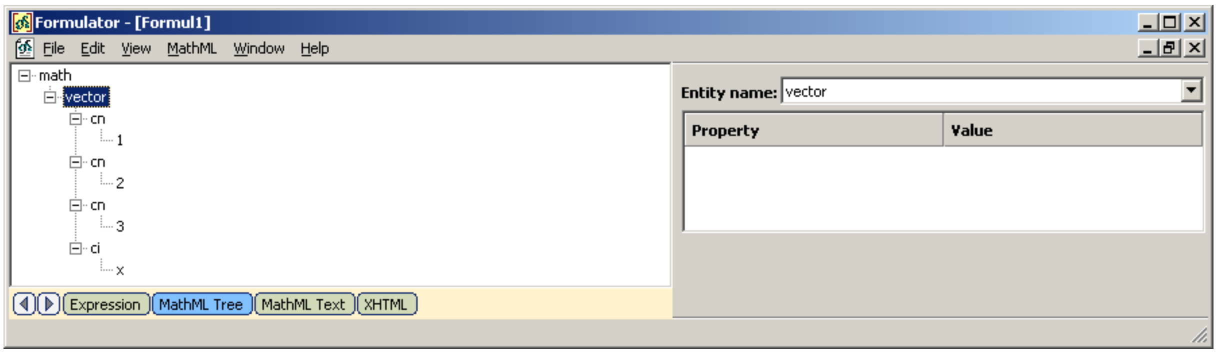
Press ‘1’, the Down arrow, ‘2’, the Down arrow, ‘3’, the Down arrow, ‘x’.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

$$\begin{pmatrix} 1 \\ 2 \\ 3 \\ x \end{pmatrix}$$

Navigation: line - content MathML; parent node - "vector" element   Size: Regular   Style: Math   Zoom: 200 %

See results:


Formulator - [Formul1]
File Edit View MathML Window Help
Expression MathML Tree MathML Text XHTML

```
math
  vector
    cn
      1
    cn
      2
    cn
      3
    ci
      x
```

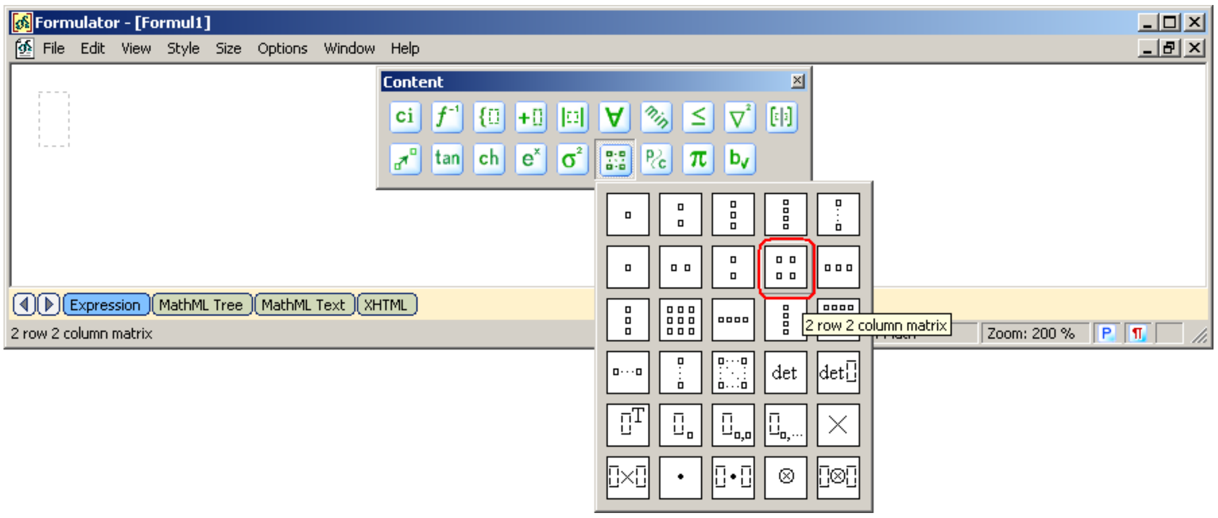
Entity name: vector

Property	Value

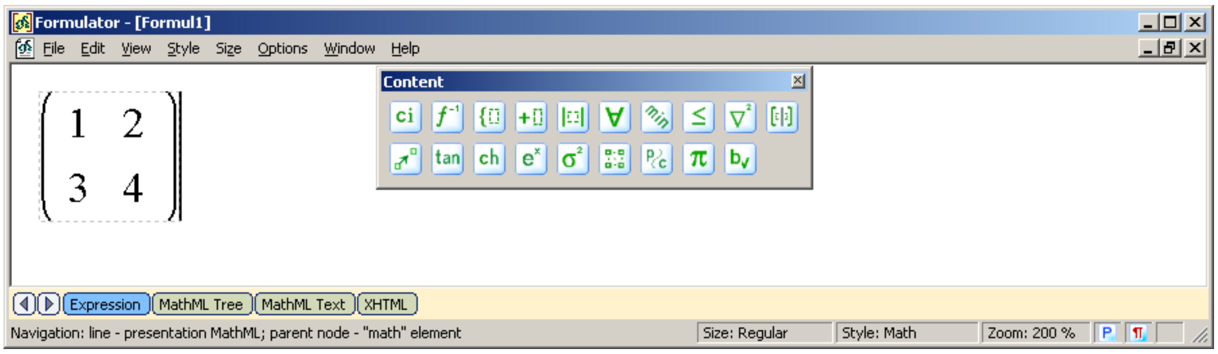
The next example show how to construct a matrix and use the “selector” element.

```
<apply>
  <selector/>
  <matrix>
    <matrixrow>
      <cn> 1 </cn> <cn> 2 </cn>
    </matrixrow>
    <matrixrow>
      <cn> 3 </cn> <cn> 4 </cn>
    </matrixrow>
  </matrix>
  <cn> 1 </cn>
```

```
</apply>
```

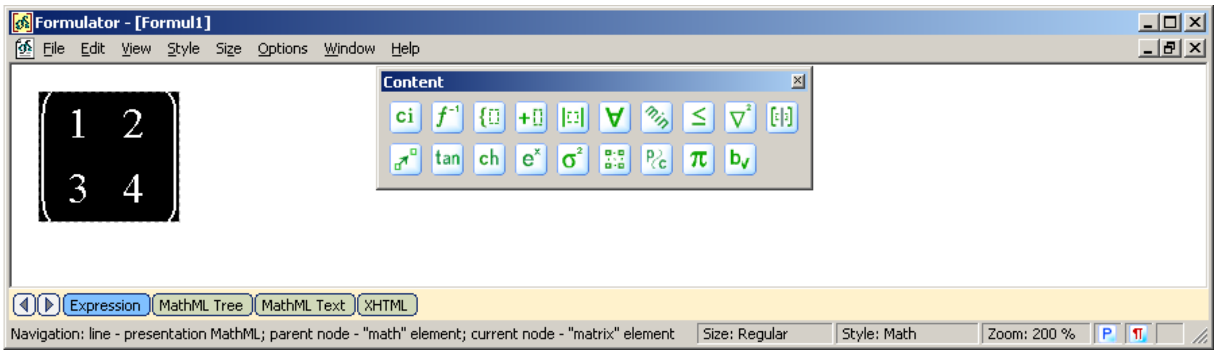

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
2 row 2 column matrix
Expression MathML Tree MathML Text XHTML
Zoom: 200 %

Press '1', the Right arrow, '2', the Right arrow, '3', the Right arrow, '4', the Right arrow.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

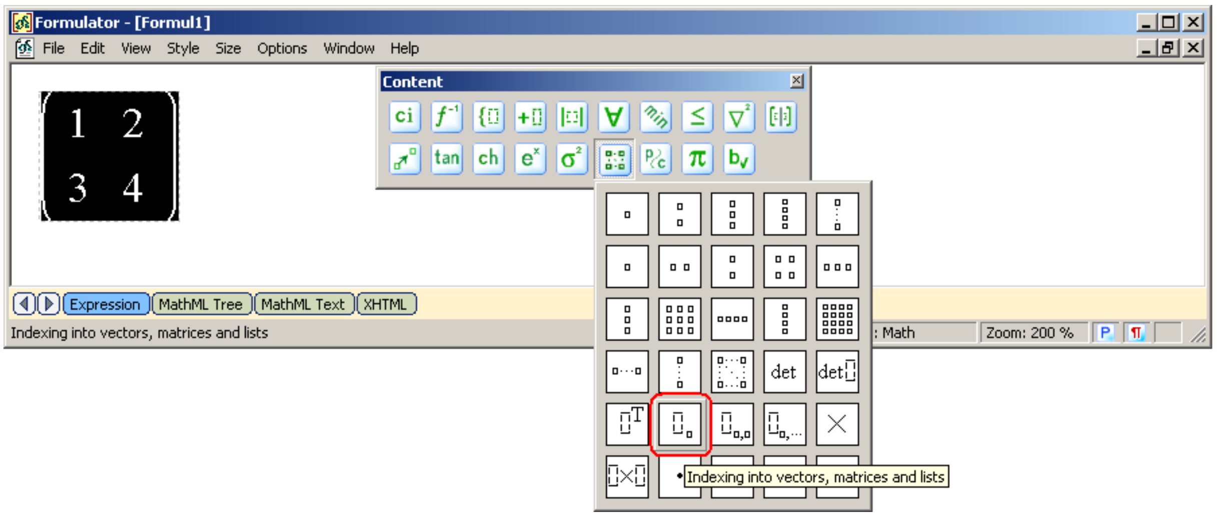
Navigation: line - presentation MathML; parent node - "math" element

Select the whole matrix.

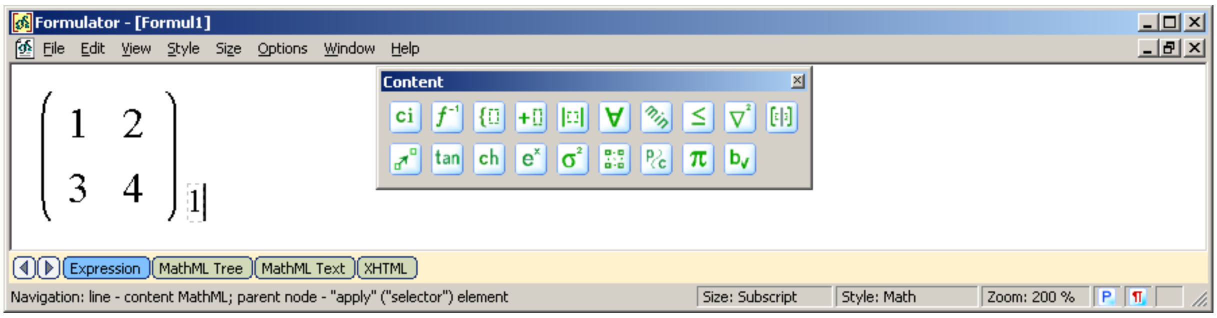

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

Navigation: line - presentation MathML; parent node - "math" element; current node - "matrix" element

Insert the "separator" element.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Indexing into vectors, matrices and lists
Zoom: 200 %

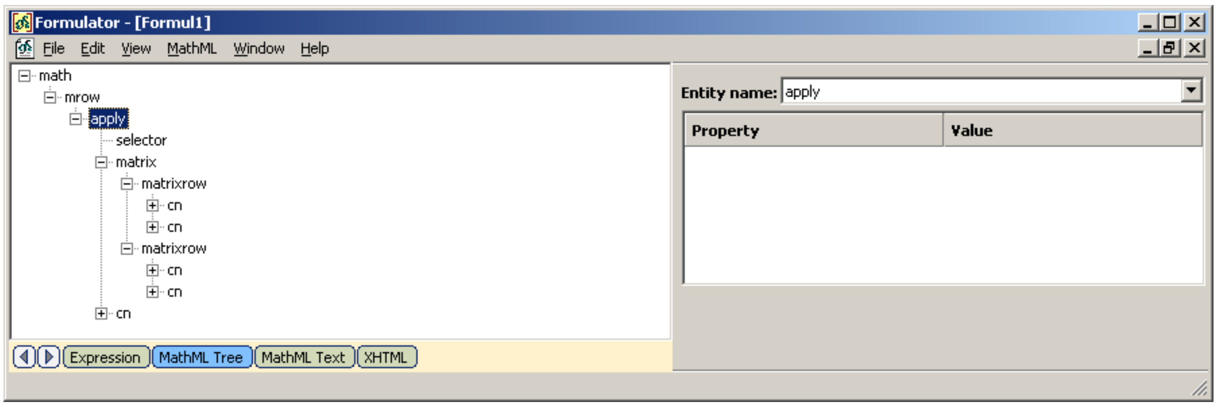
Press the Right arrow, ‘1’.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

Navigation: line - content MathML; parent node - "apply" ("selector") element

Size: Subscript Style: Math Zoom: 200 %

See the resulting MathML tree:


Formulator - [Formul1]
File Edit View MathML Window Help
Expression MathML Tree MathML Text XHTML

```
math
  mrow
    apply
      selector
      matrix
        matrixrow
          cn
          cn
        matrixrow
          cn
          cn
      cn
```

Entity name: apply

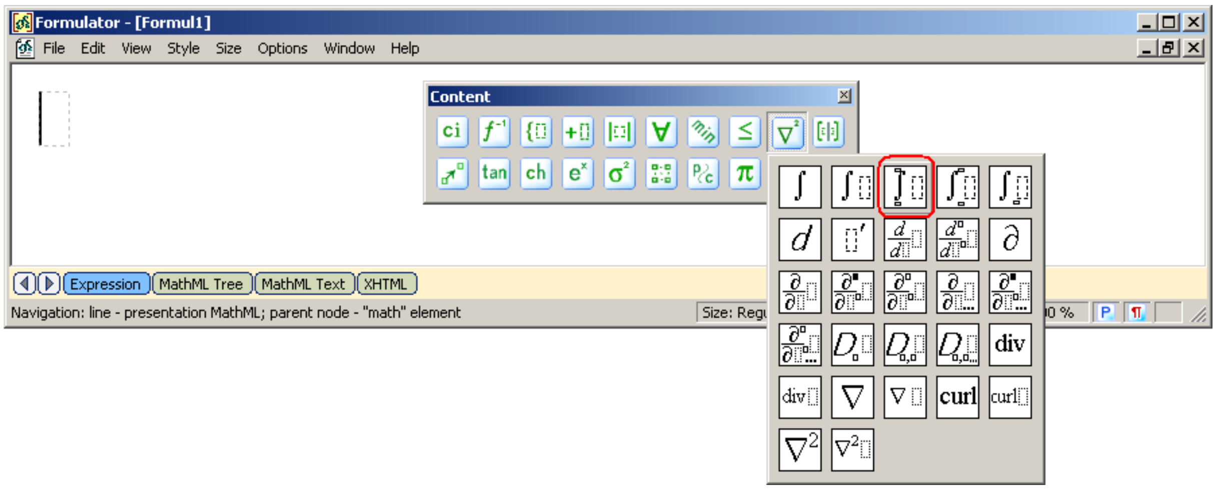
Property	Value

Calculus and Vector Calculus: Integral and Differentiation

1. Example 4.4.5.1.2 from the W3C MathML Recommendation): the int element:

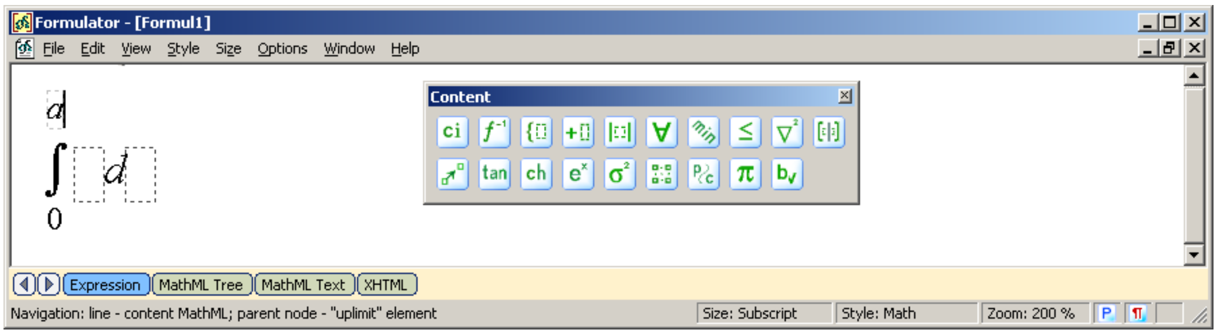
```
<apply>
  <int/>
  <bvar><ci> x </ci></bvar>
  <lowlimit><cn> 0 </cn></lowlimit>
  <uplimit><ci> a </ci></uplimit>
  <apply>
    <ci> f </ci>
    <ci> x </ci>
  </apply>
</apply>
```

Use the mathematical toolbar ∇².


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

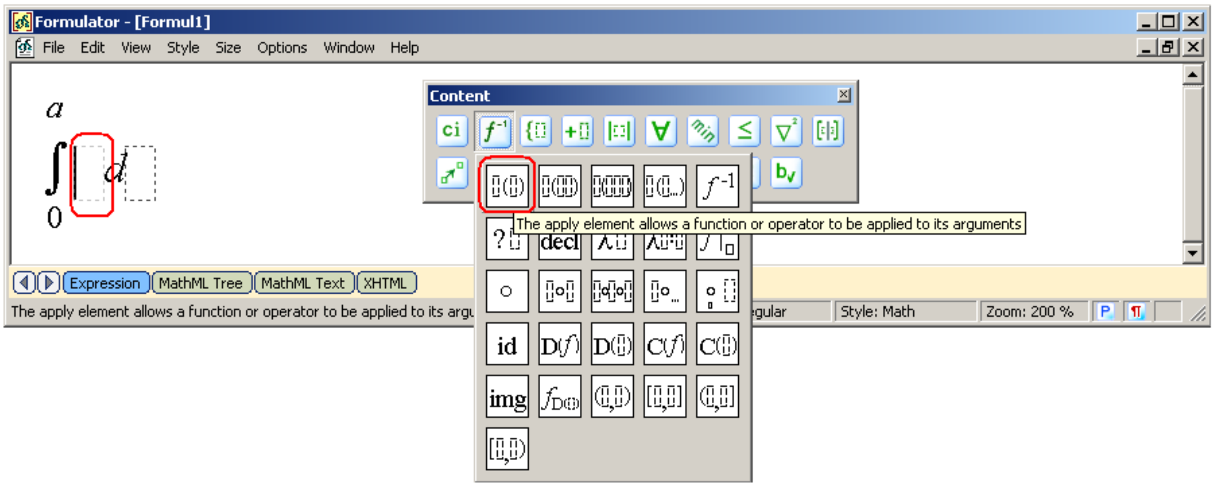
Navigation: line - presentation MathML; parent node - "math" element

Press '0', the Up arrow, 'a' (a low and upper limits of the integral).


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

Navigation: line - content MathML; parent node - "uplimit" element

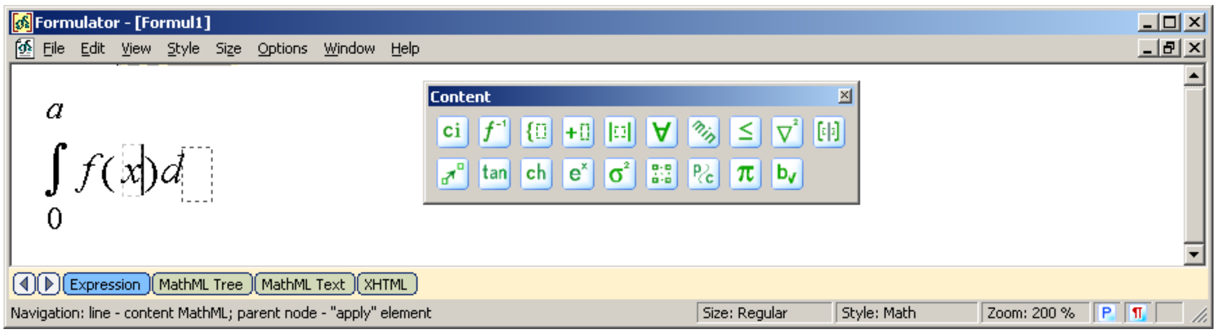
Size: Subscript Style: Math Zoom: 200 %


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML

The apply element allows a function or operator to be applied to its arguments

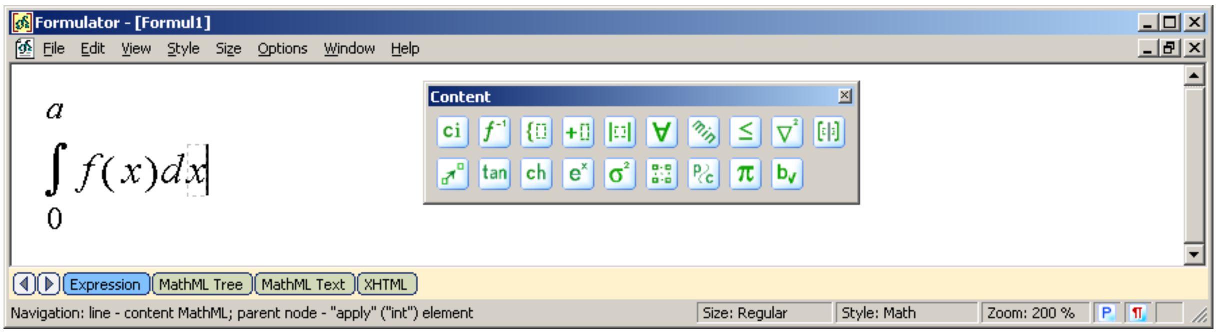
Style: Math Zoom: 200 %

Type 'f', the Right arrow, 'x'.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci tan ch π
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

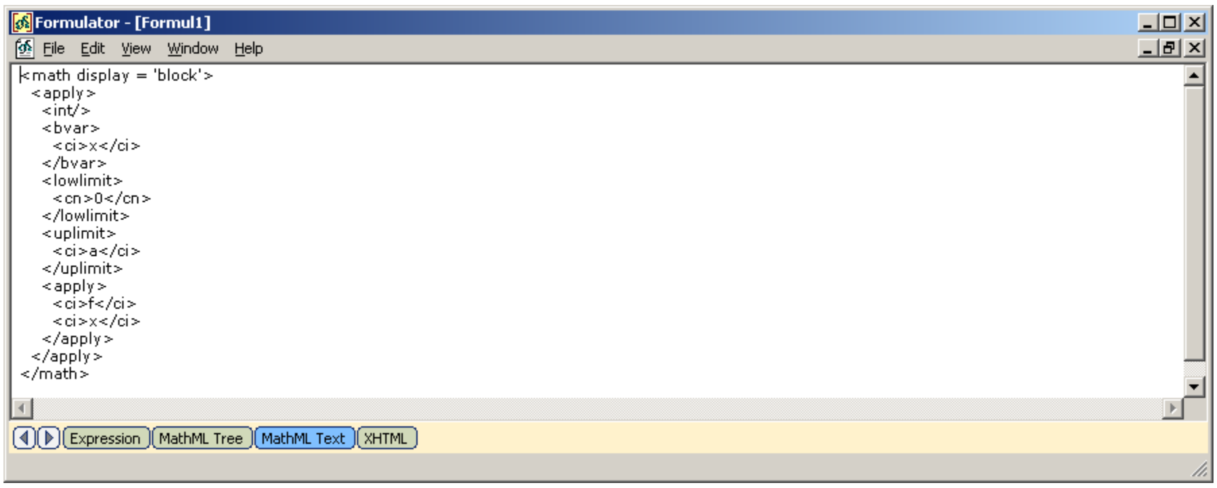
Navigation: line - content MathML; parent node - "apply" element

Press the Right arrow two times, type ‘x’ (a bounded variable).

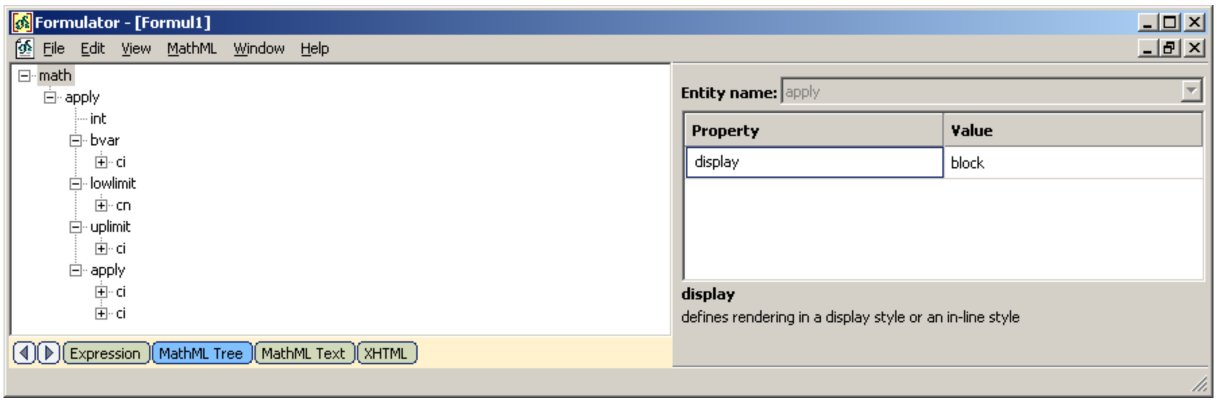

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci tan ch π
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

Navigation: line - content MathML; parent node - "apply" ("int") element

See results:


Formulator - [Formul1]
File Edit View Window Help
Expression MathML Tree MathML Text XHTML

```
<math display = 'block'>
 <apply>
  <int/>
  <bvar>
   <ci>x</ci>
  </bvar>
  <lowlimit>
   <cn>0</cn>
  </lowlimit>
  <uplimit>
   <ci>a</ci>
  </uplimit>
  <apply>
   <ci>f</ci>
   <ci>x</ci>
  </apply>
 </apply>
</math>
```


Formulator - [Formul1]
File Edit View MathML Window Help
Expression MathML Tree MathML Text XHTML

```
math
  apply
    int
    bvar
      ci
    lowlimit
      cn
    uplimit
      ci
    apply
      ci
      ci
```

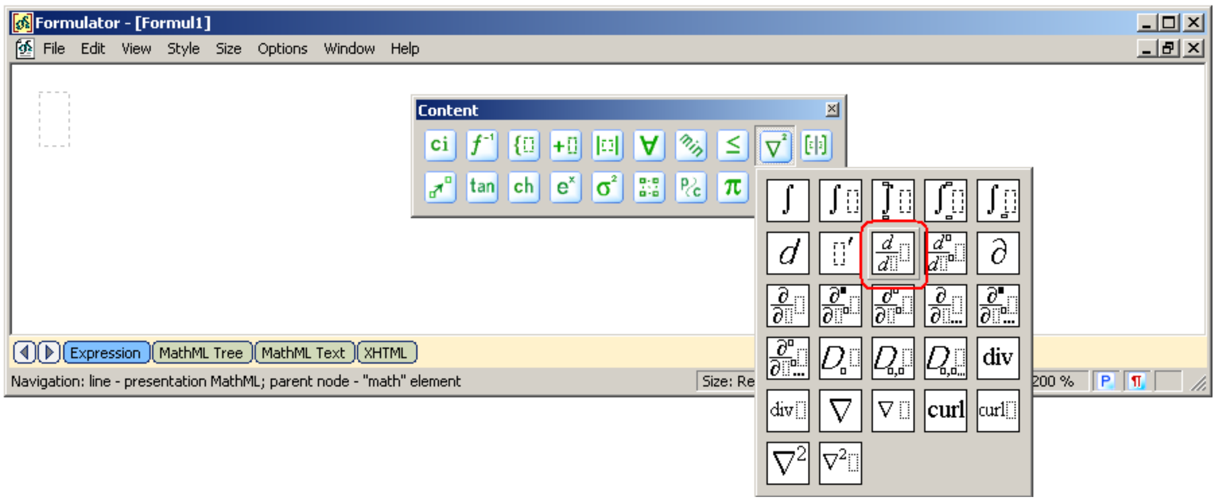
Entity name: apply

Property	Value
display	block

display
defines rendering in a display style or an in-line style

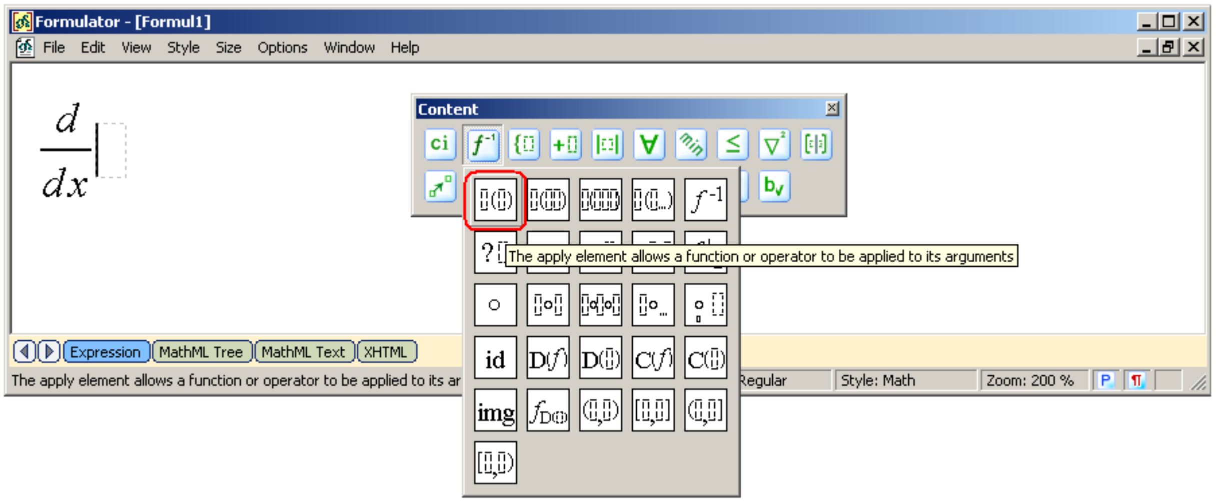
2. Example 4.4.5.2.2 from the W3C MathML Recommendation): the diff element:

```
<apply>
  <diff/>
  <bvar><ci> x </ci></bvar>
  <apply><ci type="function"> f </ci>
    <ci> x </ci>
  </apply>
</apply>
```



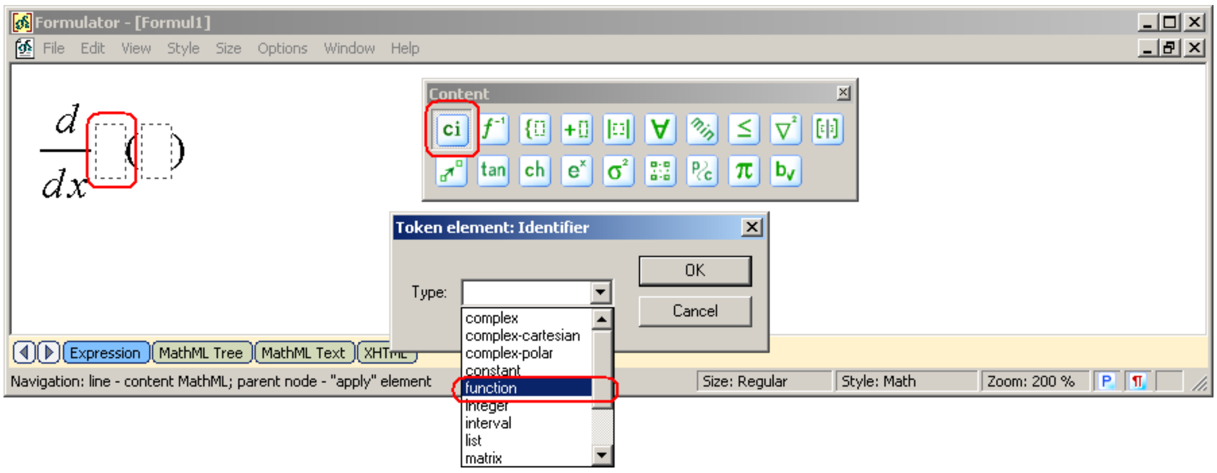
Navigation: line - presentation MathML; parent node - "math" element

Type ‘x’ (a name of the bounded variable), press the Right arrow.

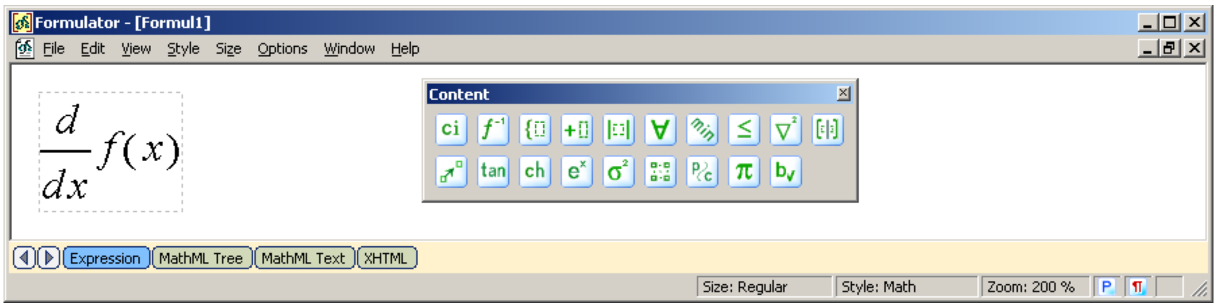


The apply element allows a function or operator to be applied to its arguments

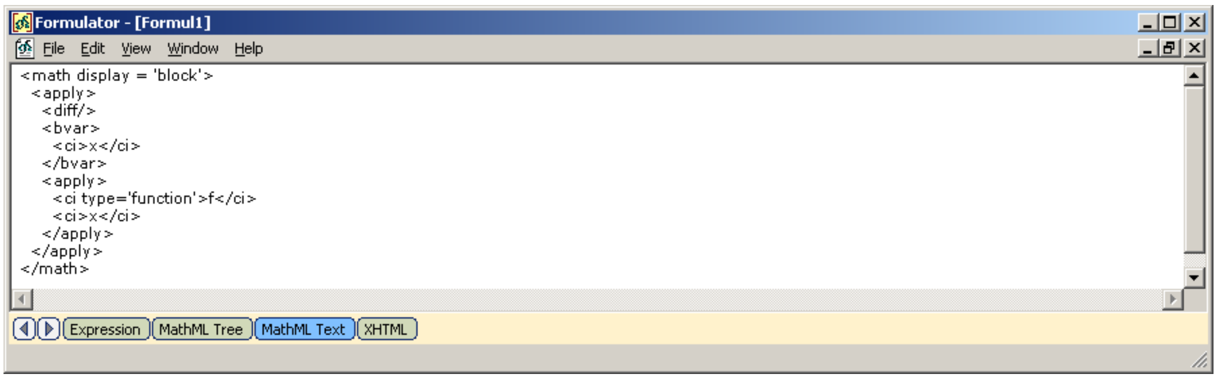
Insert the “ci” element of type “function”; name it “f”.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci tan ch
Token element: Identifier
Type:
OK
Cancel
complex
complex-cartesian
complex-polar
constant
function
integer
interval
list
matrix
Expression MathML Tree MathML Text XHTML
Navigation: line - content MathML; parent node - "apply" element
Size: Regular Style: Math Zoom: 200 %

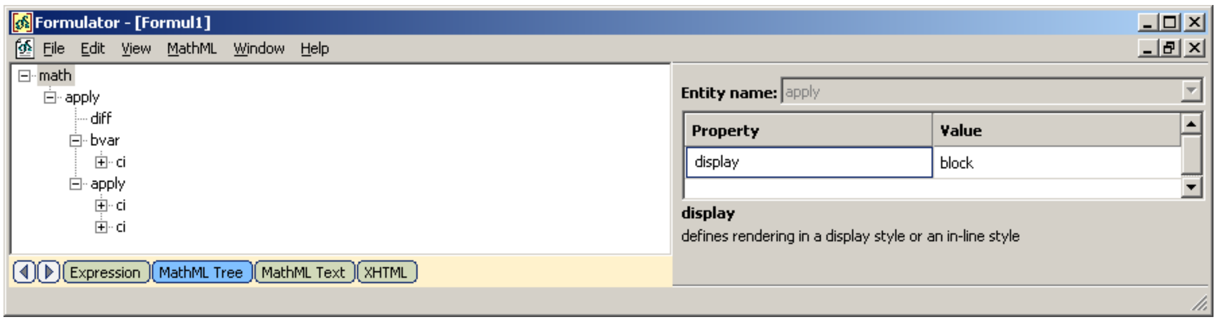
Type ‘f’, press the Right arrow two times (the first in order to get out of the “ci” node, the second to get to the second input slot of the “apply” element). Type ‘x’.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular Style: Math Zoom: 200 %

See results:


Formulator - [Formul1]
File Edit View Window Help
Expression MathML Tree MathML Text XHTML

```
<math display = 'block'>
 <apply>
  <diff/>
  <bvar>
   <ci>x</ci>
  </bvar>
  <apply>
   <ci type='function'>f</ci>
   <ci>x</ci>
  </apply>
 </apply>
</math>
```

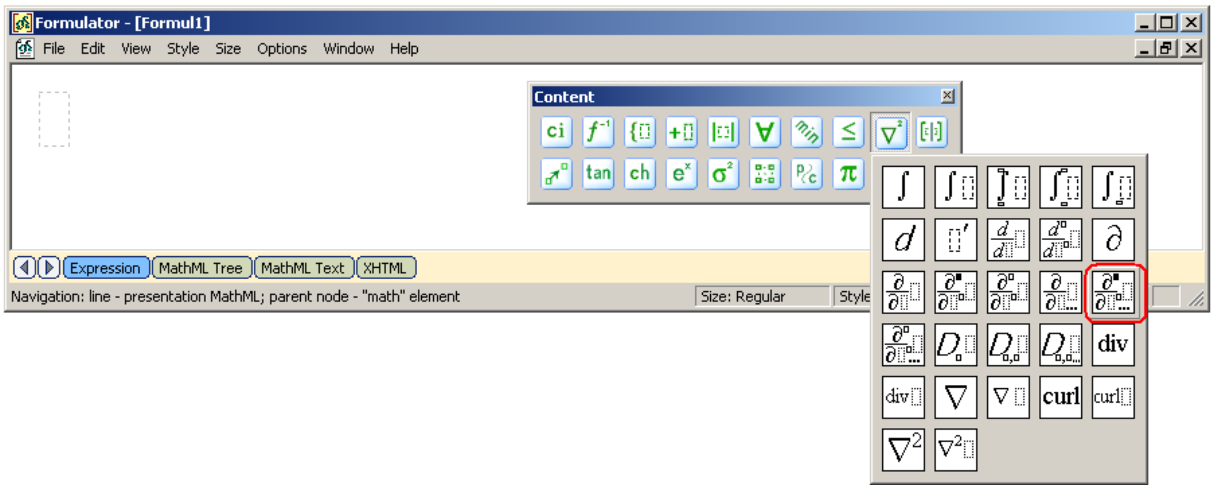

Formulator - [Formul1]
File Edit View MathML Window Help
math
apply
diff
bvar
ci
apply
ci
ci
Entity name: apply
Expression MathML Tree MathML Text XHTML

Property	Value
display	block

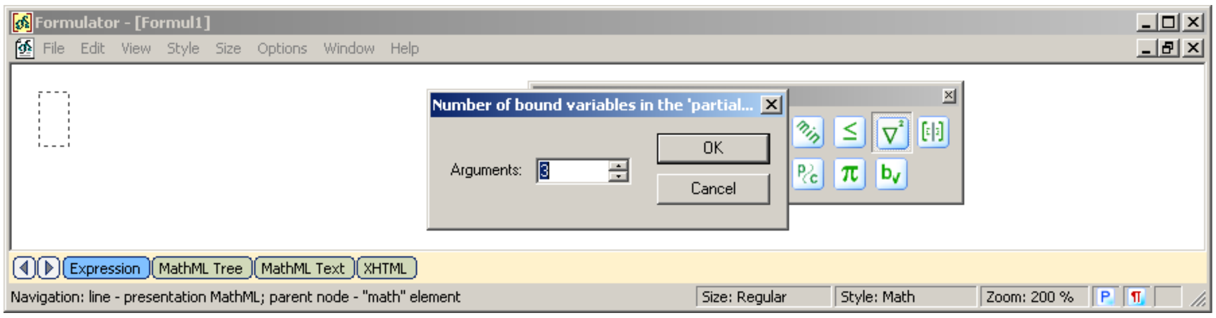
display
defines rendering in a display style or an in-line style

Partial Differentiation

1. The "partialdiff" element can automatically calculate a total degree of differentiation by use of child `degree` elements. The next example demonstrates this feature.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular

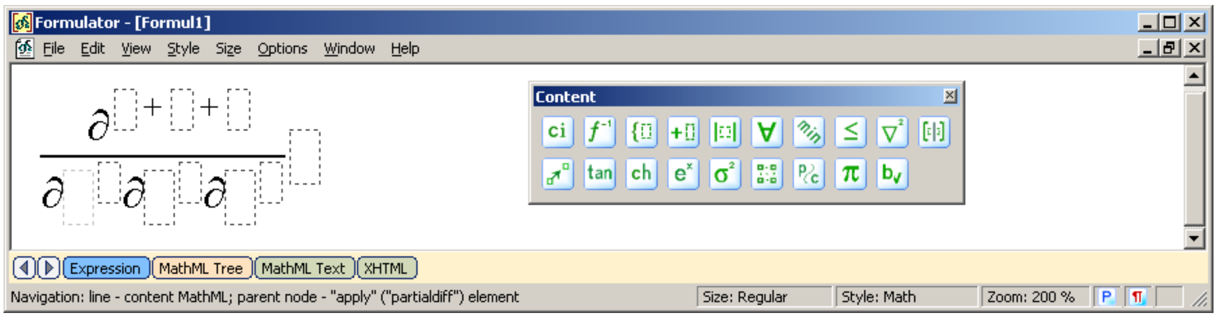
Navigation: line - presentation MathML; parent node - "math" element


Formulator - [Formul1]
File Edit View Style Size Options Window Help
OK
Cancel
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

Number of bound variables in the 'partial...

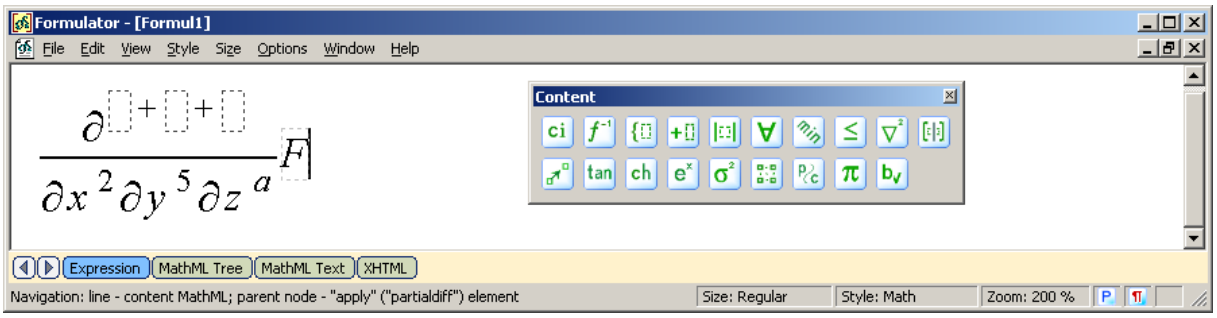
Arguments: 3

Navigation: line - presentation MathML; parent node - "math" element


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

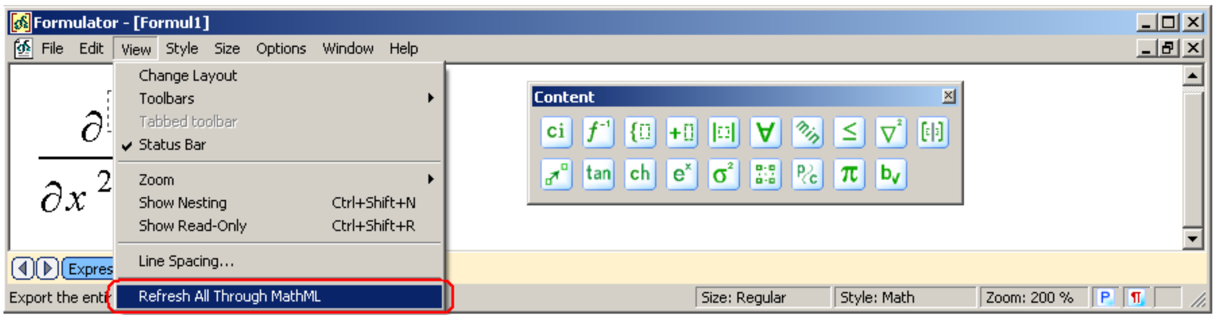
Navigation: line - content MathML; parent node - "apply" ("partialdiff") element

Input names of bounded variables and choose a numeric and symbolic values of their degree.

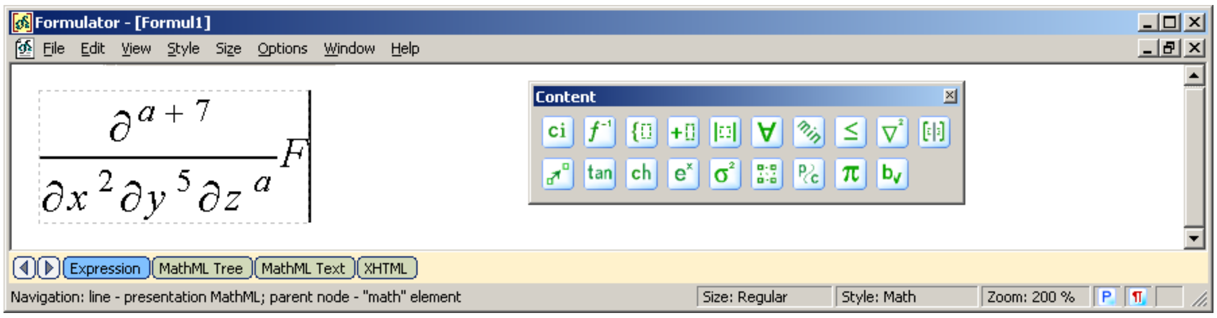

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

Navigation: line - content MathML; parent node - "apply" ("partialdiff") element

Now a total degree of differentiation can be calculated if we refresh the text through MathML.

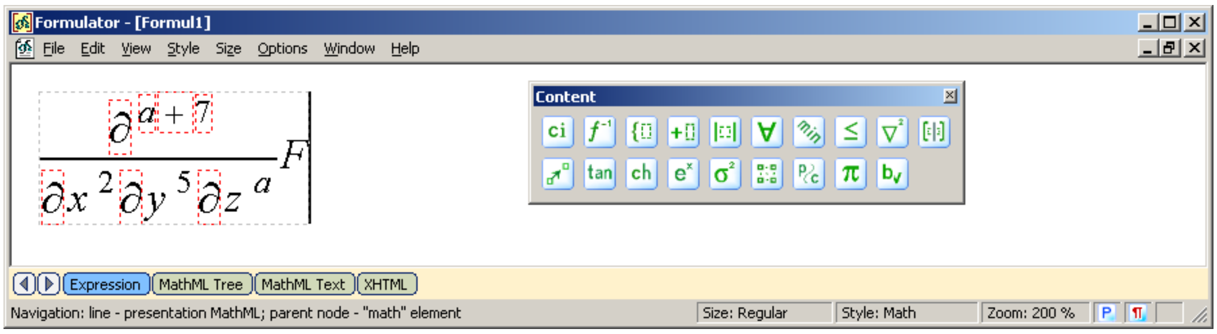


See results:

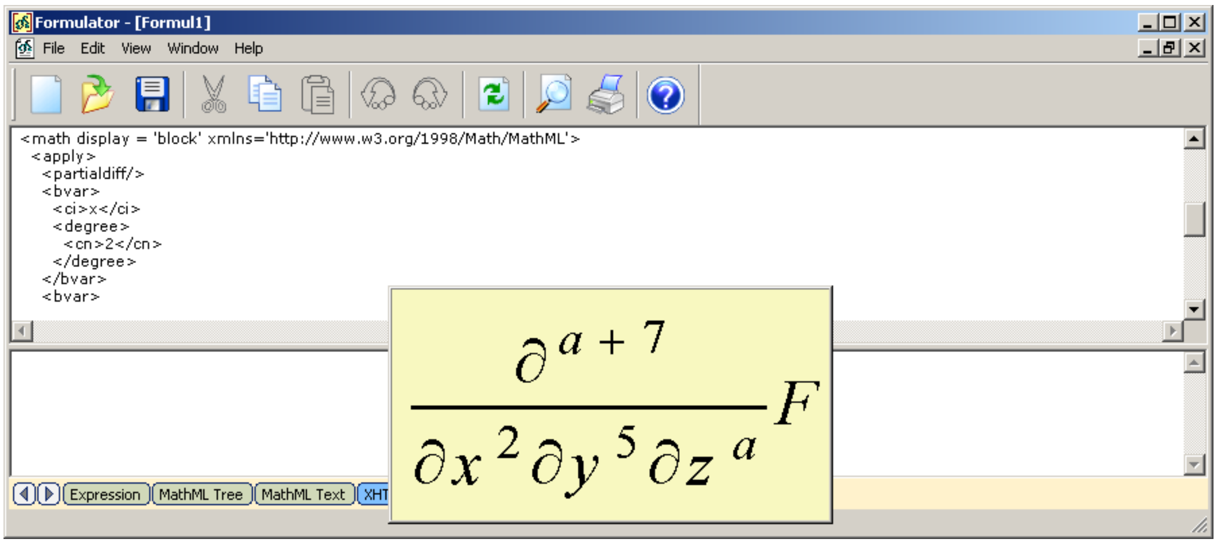

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression
MathML Tree
MathML Text
XHTML
Navigation: line - presentation MathML; parent node - "math" element
Size: Regular
Style: Math
Zoom: 200 %

$$\frac{\partial^{a+7}}{\partial x^2 \partial y^5 \partial z^a} F$$

Turn on the "Show Read-Only" option from the "View" menu to see which areas can't be edited in the created formula. The next figure suggests that automatically detected total degree of the "partialdiff" element can't be edited (dashed red line around elements).


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression
MathML Tree
MathML Text
XHTML
Navigation: line - presentation MathML; parent node - "math" element
Size: Regular
Style: Math
Zoom: 200 %

See the same results on the "XHTML" page of MathML Weaver, using the 'Zoom' feature from the context menu.

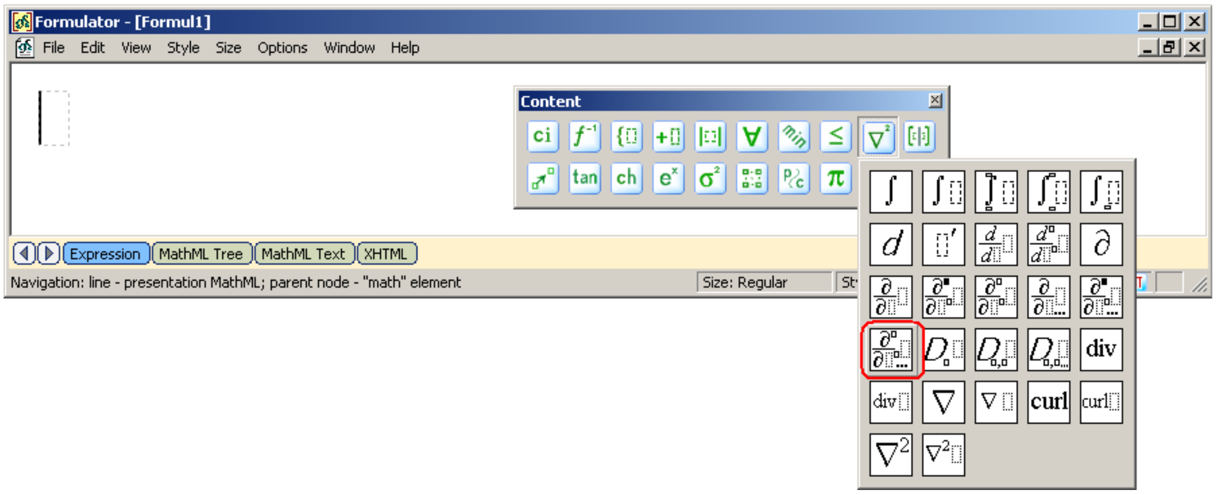

Formulator - [Formul1]
File Edit View Window Help
Expression MathML Tree MathML Text XHT

```
<math display = 'block' xmlns='http://www.w3.org/1998/Math/MathML'>
 <apply>
  <partialdiff/>
  <bvar>
   <ci>x</ci>
   <degree>
    <cn>2</cn>
   </degree>
  </bvar>
  <bvar>
```

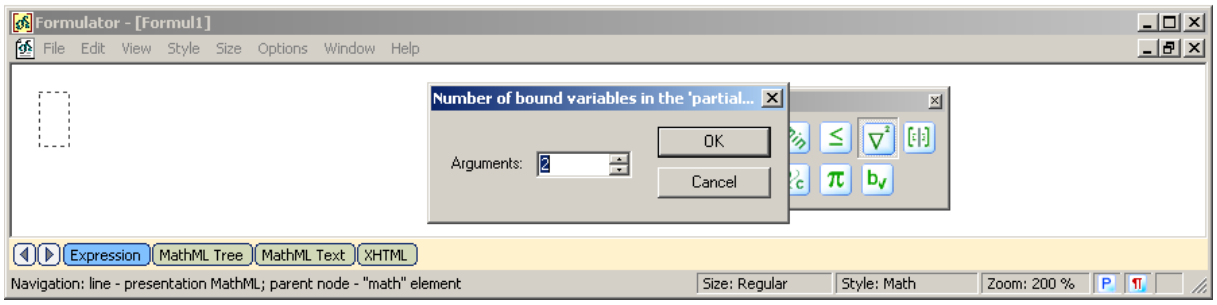
$$\frac{\partial^{a+7}}{\partial x^{2}\partial y^{5}\partial z^{a}}F$$

2. Example 4.4.5.3.2 from the W3C MathML Recommendation): the partialdiff element:

```
<apply><partialdiff/>
 <bvar><ci> x </ci><degree><ci> m </ci></degree></bvar>
 <bvar><ci> y </ci><degree><ci> n </ci></degree></bvar>
 <degree><ci> k </ci></degree>
 <apply><ci type="function"> f </ci>
  <ci> x </ci>
  <ci> y </ci>
 </apply>
</apply>
```


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Regular

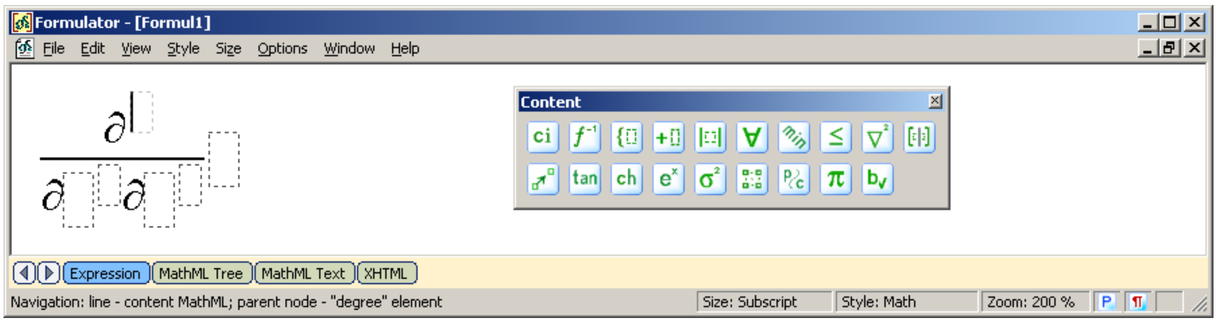
Navigation: line - presentation MathML; parent node - "math" element


Formulator - [Formul1]
File Edit View Style Size Options Window Help
OK
Cancel
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

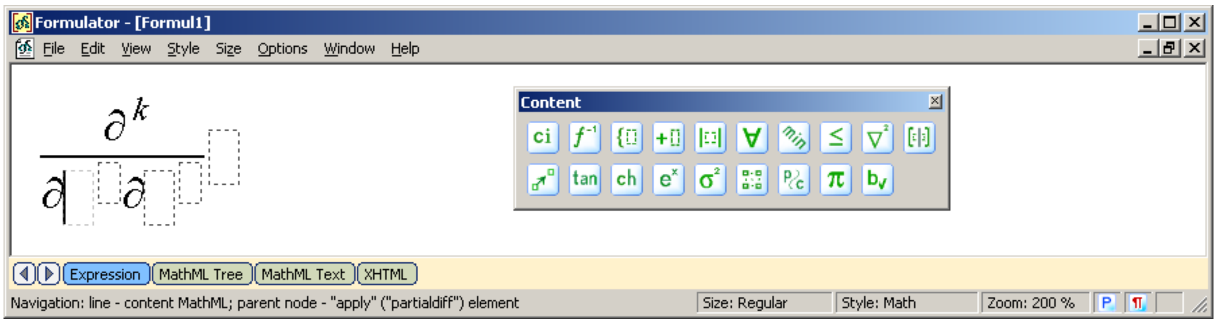
Number of bound variables in the 'partial...

Arguments: 2

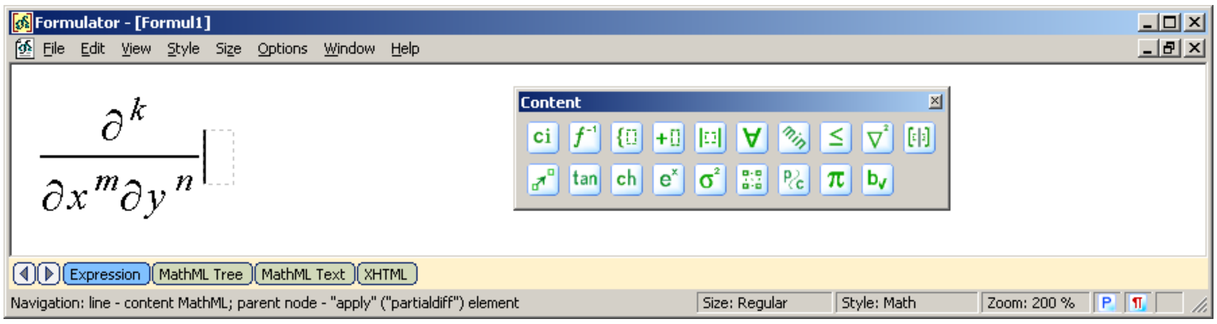
Navigation: line - presentation MathML; parent node - "math" element

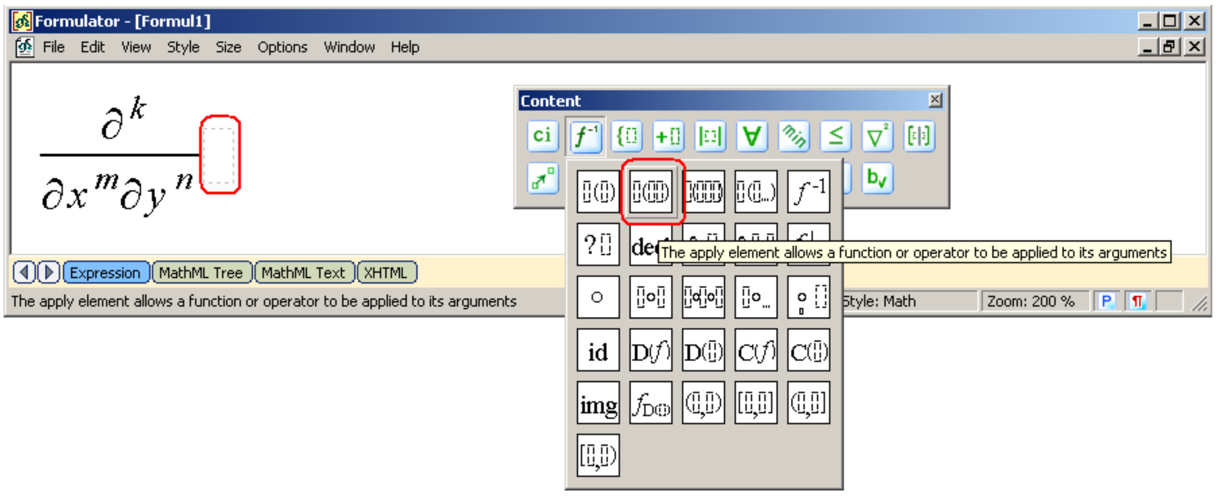

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Navigation: line - content MathML; parent node - "degree" element
Size: Subscript Style: Math Zoom: 200 %

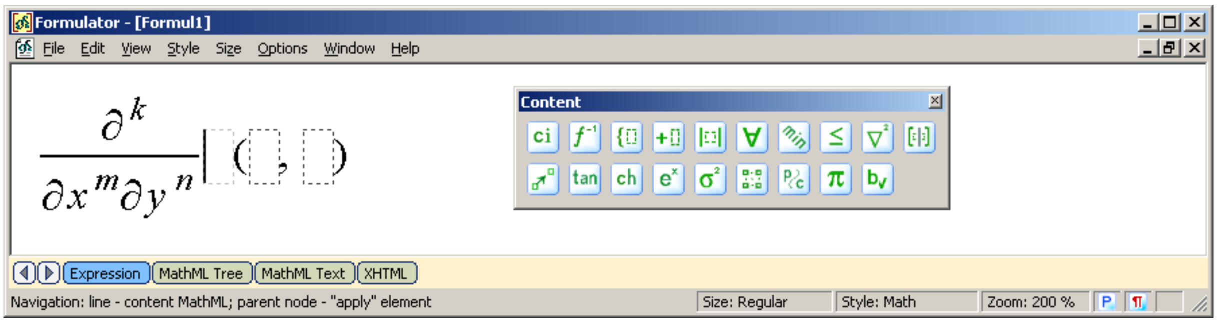
Input a total degree of differentiation by typing 'k'. Press the Right arrow to start inputting bounded variables.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Navigation: line - content MathML; parent node - "apply" ("partialdiff") element
Size: Regular Style: Math Zoom: 200 %

Press 'x', the Right arrow, 'm', the Right arrow, 'y', the Right arrow, 'n', the Right arrow.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Navigation: line - content MathML; parent node - "apply" ("partialdiff") element
Size: Regular Style: Math Zoom: 200 %

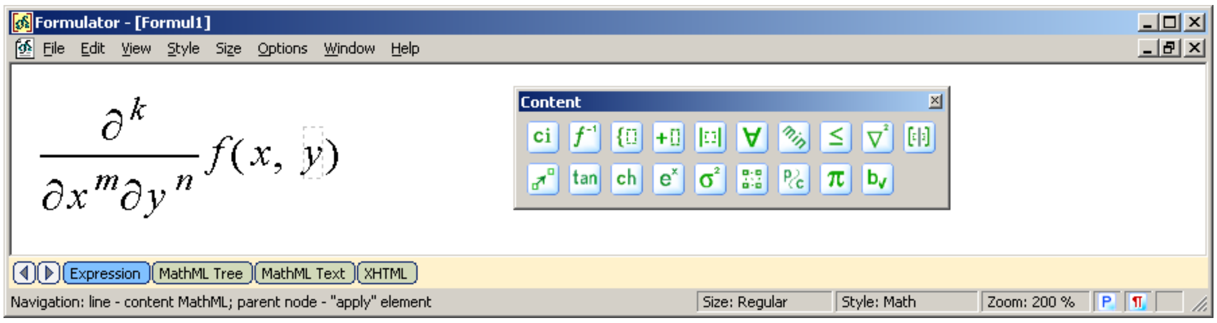

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
The apply element allows a function or operator to be applied to its arguments
Style: Math Zoom: 200 %


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci tan ch π
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

$$\frac{\partial^k}{\partial x^m \partial y^n}\,\boxed{}(\boxed{},\boxed{})$$

Navigation: line - content MathML; parent node - "apply" element

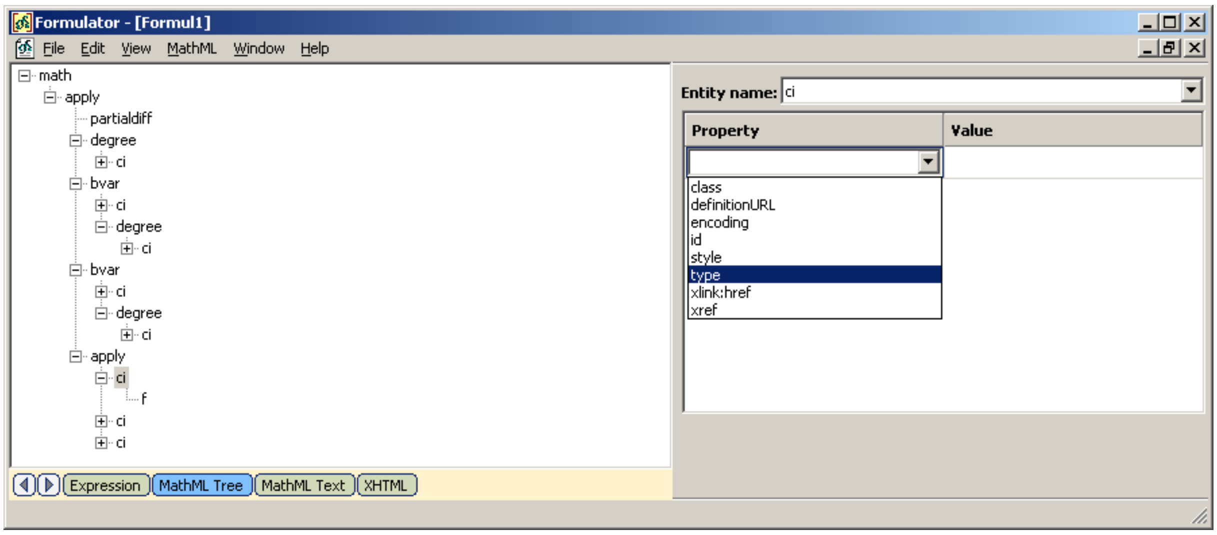
Press ‘f’, the Right arrow, ‘x’, the Right arrow, ‘y’.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
ci tan ch π
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

$$\frac{\partial^k}{\partial x^m \partial y^n} f(x, y)$$

Navigation: line - content MathML; parent node - "apply" element

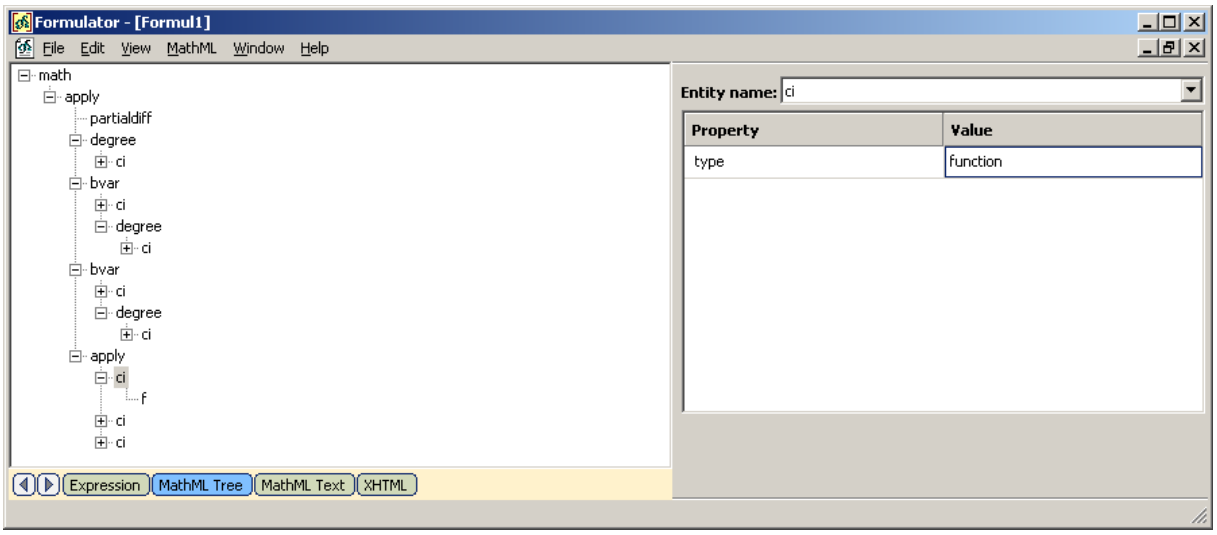
Switch to the “MathML Tree” and add the “type” attribute to the function ‘f’.


Formulator - [Formul1]
File Edit View MathML Window Help
Expression MathML Tree MathML Text XHTML

```
math
  apply
    partialdiff
    degree
      ci
    bvar
      ci
      degree
        ci
    bvar
      ci
      degree
        ci
    apply
      ci
        f
      ci
      ci
```

Entity name: ci

Property	Value
class	
definitionURL	
encoding	
id	
style	
type	
xlink:href	
xref	

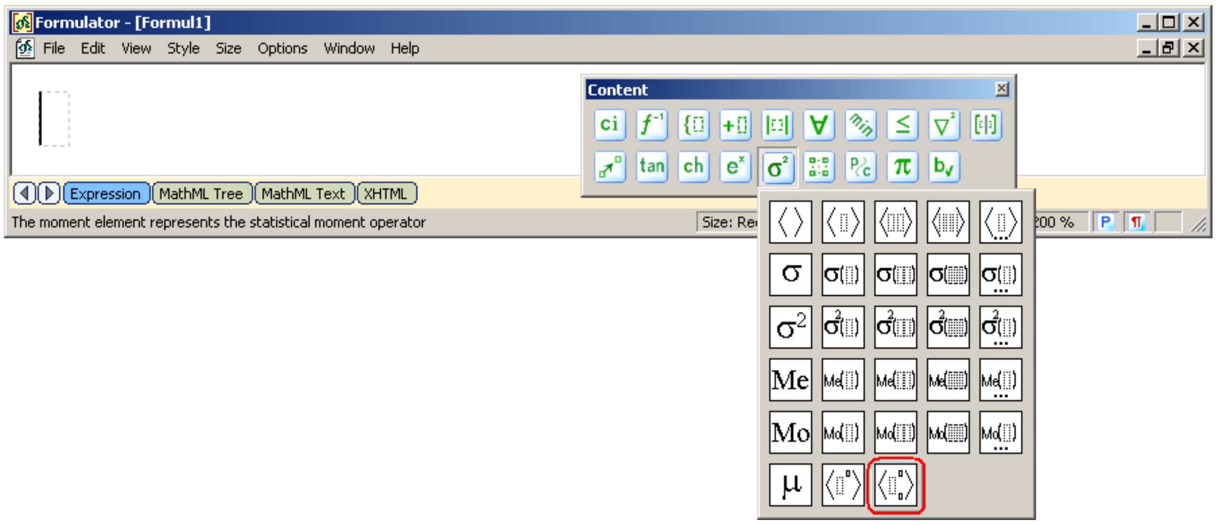


Property	Value
type	function

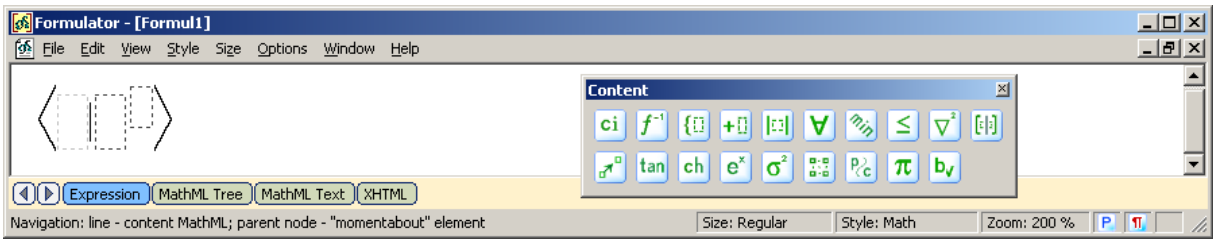
Statistics

Lets' consider an example 4.4.9.6.2 from the W3C MathML Recommendation): the partialdiff element:

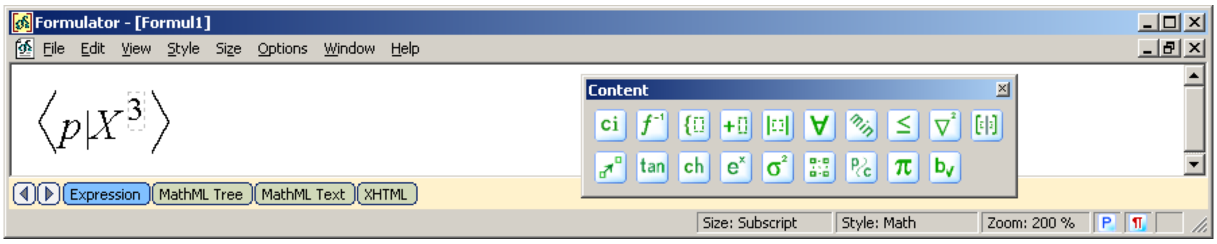
```
<apply>
  <moment/>
  <degree><cn> 3 </cn></degree>
  <momentabout>
    <ci> p </ci>
  </momentabout>
  <ci> X </ci>
</apply>
```



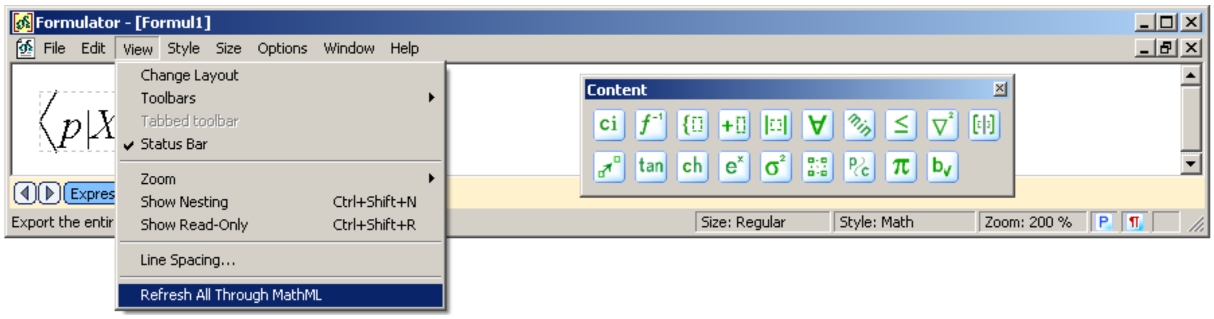
The moment element represents the statistical moment operator

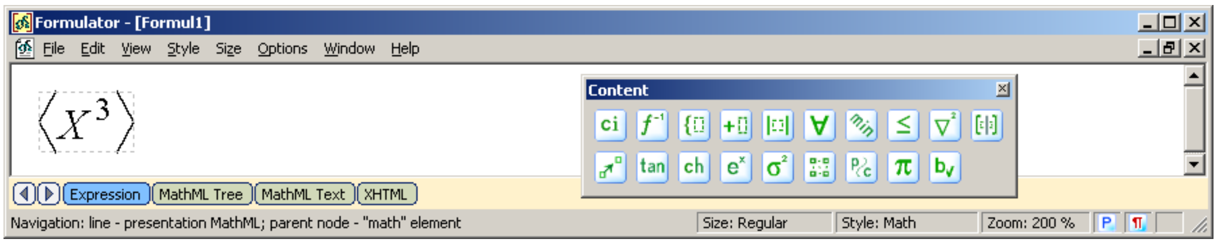


Navigation: line - content MathML; parent node - "momentabout" element



Get rid of the input slot for the "momentabout" element (that should not be finally rendered).



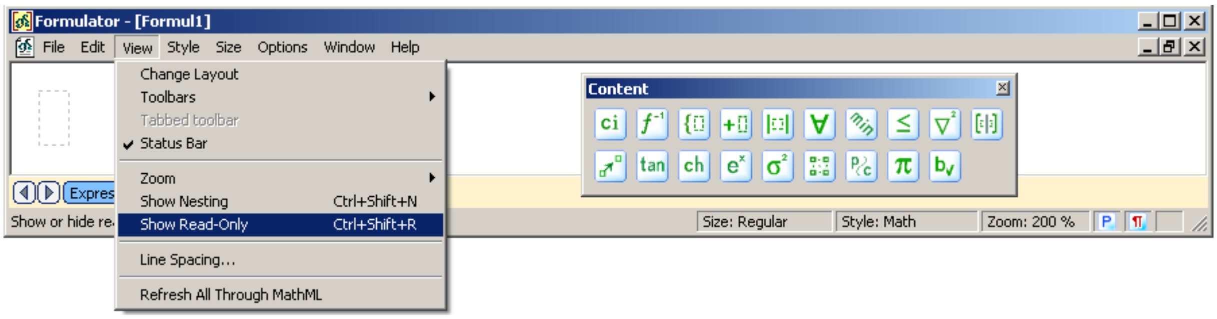

Formulator - [Formul1]
File Edit View Style Size Options Window Help
$\langle X^3\rangle$
Content
Expression MathML Tree MathML Text XHTML
Size: Regular
Style: Math
Zoom: 200 %

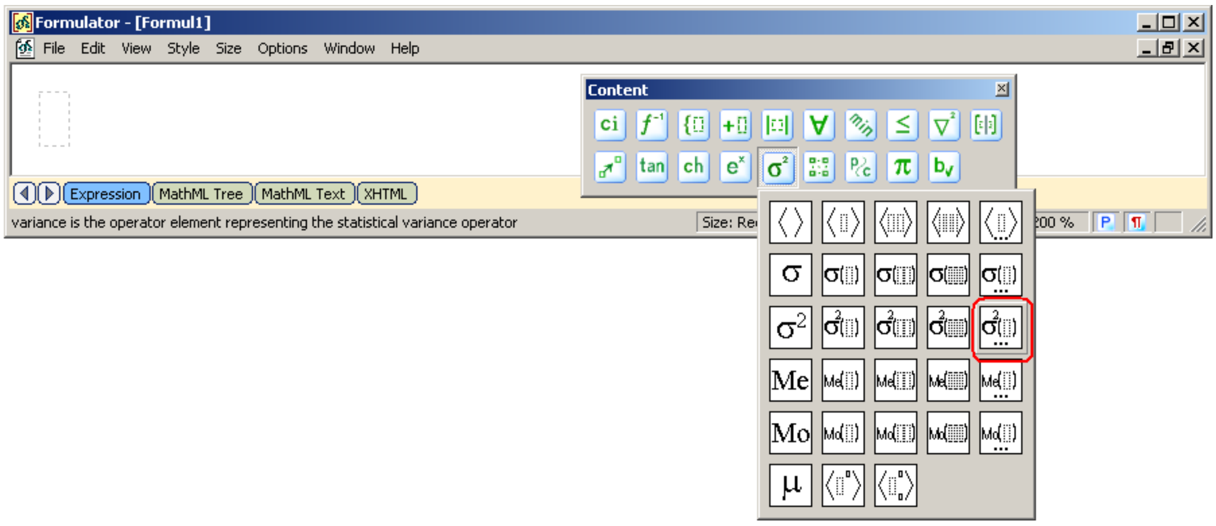
Navigation: line - presentation MathML; parent node - "math" element

Read-only elements

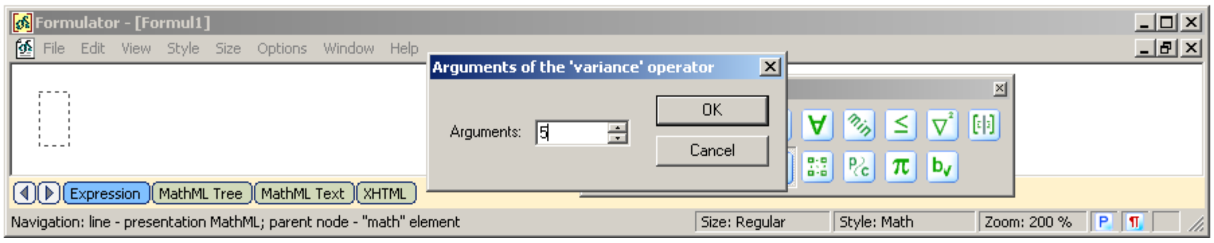
Create the "variance" element with 5 arguments and see which elements are read-only in the visual editing form.

Turn on the "Show Read-Only" option from the "View" menu.




Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Expression MathML Tree MathML Text XHTML
Size: Re
200 %

variance is the operator element representing the statistical variance operator


Formulator - [Formul1]
File Edit View Style Size Options Window Help
OK
Cancel
Expression MathML Tree MathML Text XHTML

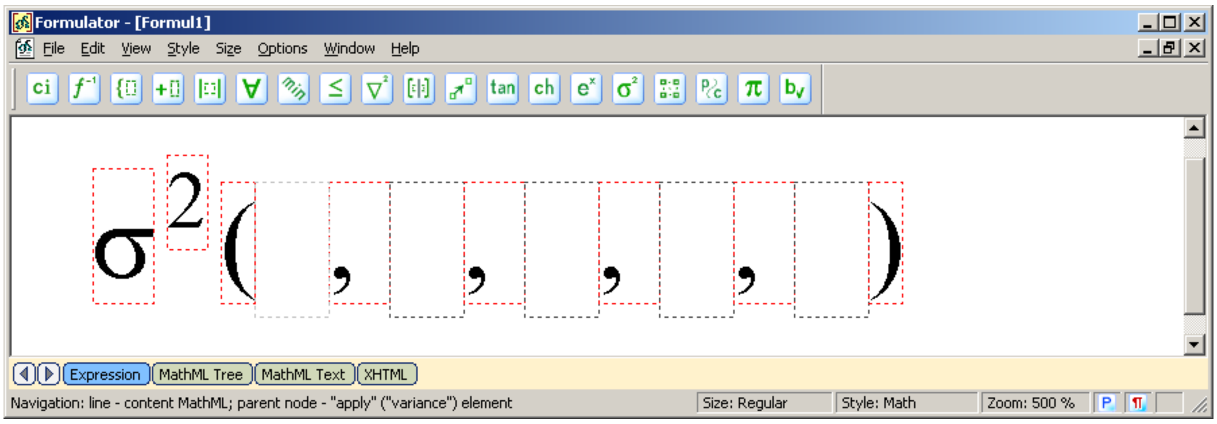
Arguments of the 'variance' operator

Arguments: 5

Navigation: line - presentation MathML; parent node - "math" element

Size: Regular Style: Math Zoom: 200 %

See how MathML Weaver make it easy to navigate through the formula by marking some MathML nodes as *read-only*.

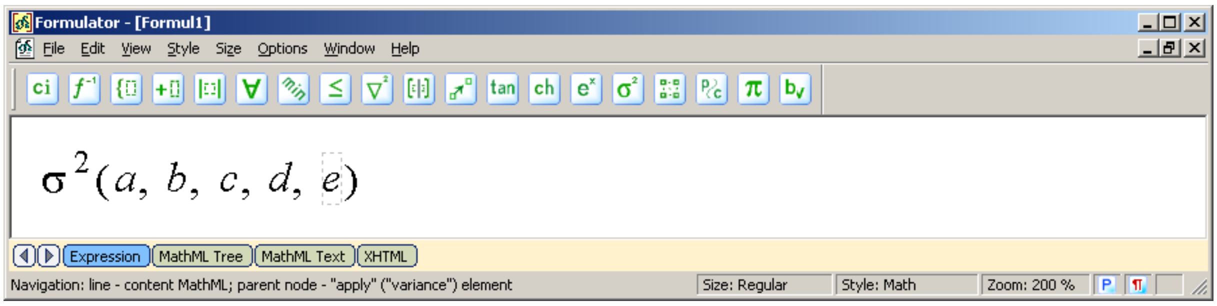

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Expression MathML Tree MathML Text XHTML

Navigation: line - content MathML; parent node - "apply" ("variance") element

Size: Regular Style: Math Zoom: 500 %

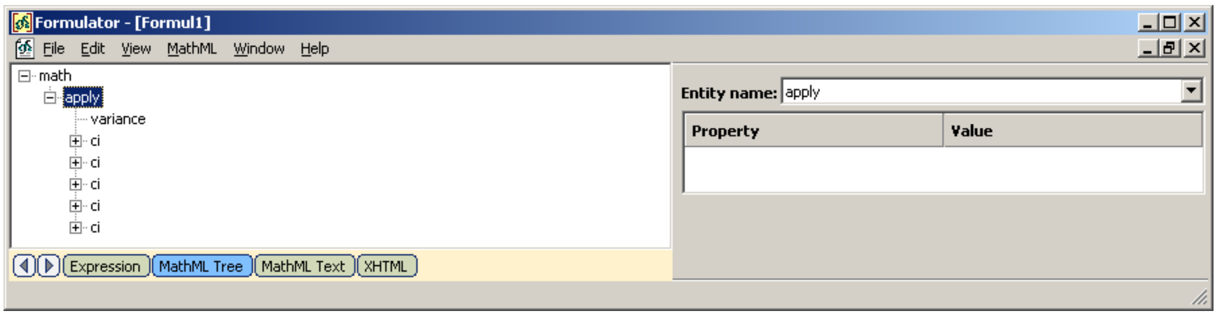
Red dashed line around an element means that this element can't be edited, because it is an essential part of a visual editing form for some element of the content markup. When a user presses navigation buttons (e.g., arrows), MathML Weaver omits visiting every element of the form, but rather chooses only not read-only elements. Due to this smart behavior a user can navigate through the MathML tree much faster, without positioning on every tree hierarchy level and every existing node.

Finish editing the "variance" element and see results.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
ci
tan
ch
π
$\sigma^2(a, b, c, d, e)$
Expression
MathML Tree
MathML Text
XHTML

Navigation: line - content MathML; parent node - "apply" ("variance") element

Size: Regular | Style: Math | Zoom: 200 %



```
math
  apply
    variance
    ci
    ci
    ci
    ci
    ci
```

Entity name: apply

Property	Value

Combining Presentation and Content Markup

MathML Weaver is able to combine Presentation and Content Markup of MathML in three ways:

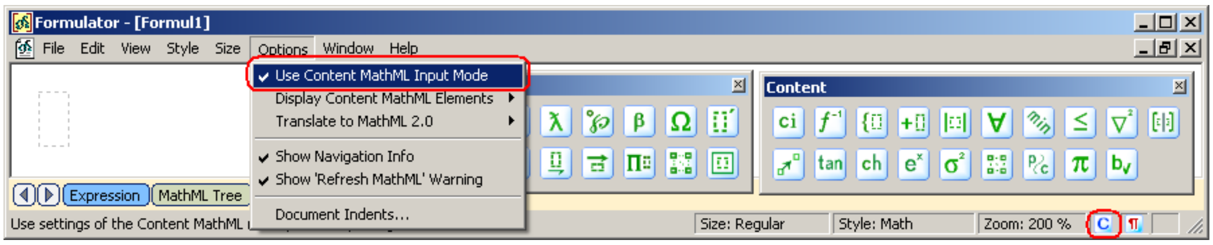
1. Presentation elements can be inserted into the input slots of editing form of the content markup. When the presence of such combining seems inappropriate by W3C Recommendation, presentation elements are wrapped into the "csymbol" an element, because MathML Weaver consider such cases as definition on-the-fly of an element in MathML whose semantics are externally defined.

2. The simplest way how content markup can be contained in presentation markup is just simultaneously using the Presentation and Content mathematical toolbars, thus embedding content markup nodes into the existing presentation hierarchy of elements. Note that the resulting expression should still have an unambiguous rendering.

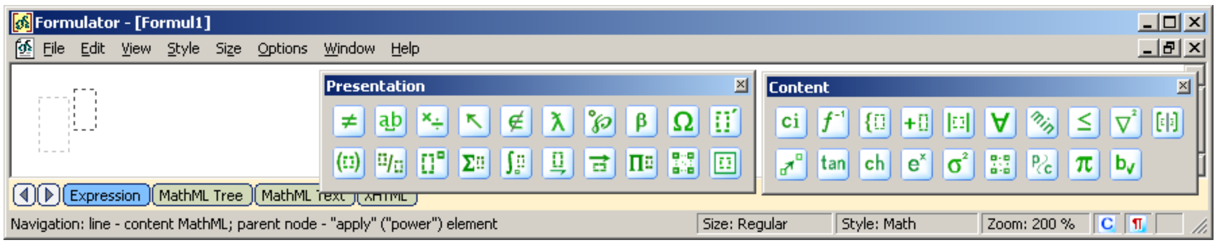
An example from the 5.2.1 section of the W3C Recommendation.

```
<mrow>
  <apply>
    <power/>
    <ci>x</ci>
    <cn>2</cn>
  </apply>
  <mo>+</mo>
  <msup>
    <mi>v</mi>
    <mn>2</mn>
  </msup>
</mrow>
```

Switch to the Content MathML input mode.

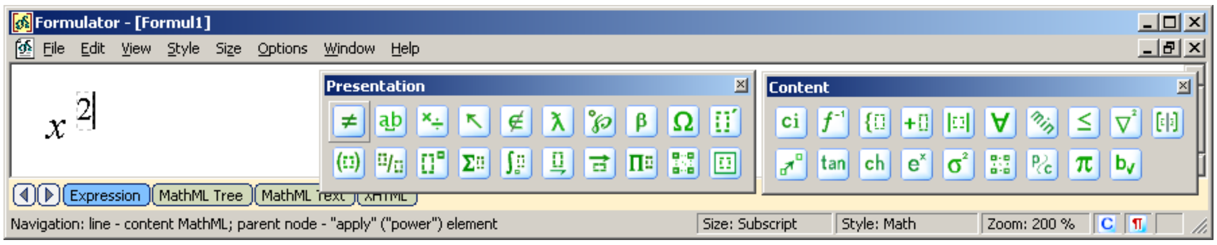

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Use Content MathML Input Mode
Display Content MathML Elements
Translate to MathML 2.0
Show Navigation Info
Show 'Refresh MathML' Warning
Document Indents...
Content
Expression MathML Tree
Use settings of the Content MathML i
Size: Regular Style: Math Zoom: 200 %

Type ‘^’ to insert the “apply” element with the <power/> operation.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Presentation
Content
Expression MathML Tree MathML Text XHTML
Size: Regular Style: Math Zoom: 200 %

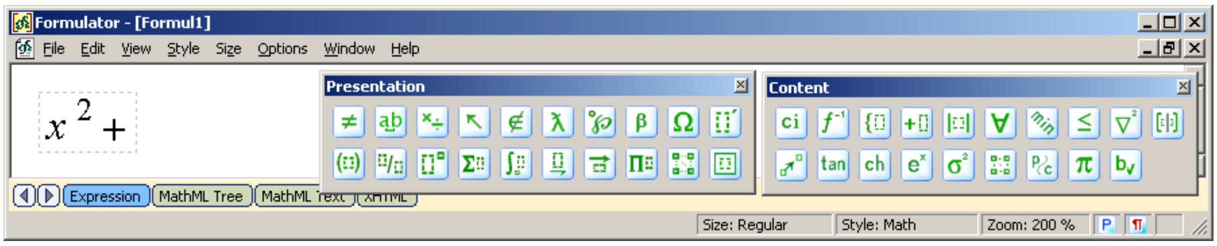
Navigation: line - content MathML; parent node - "apply" ("power") element

Press ‘x’, the Right arrow, ‘2’.

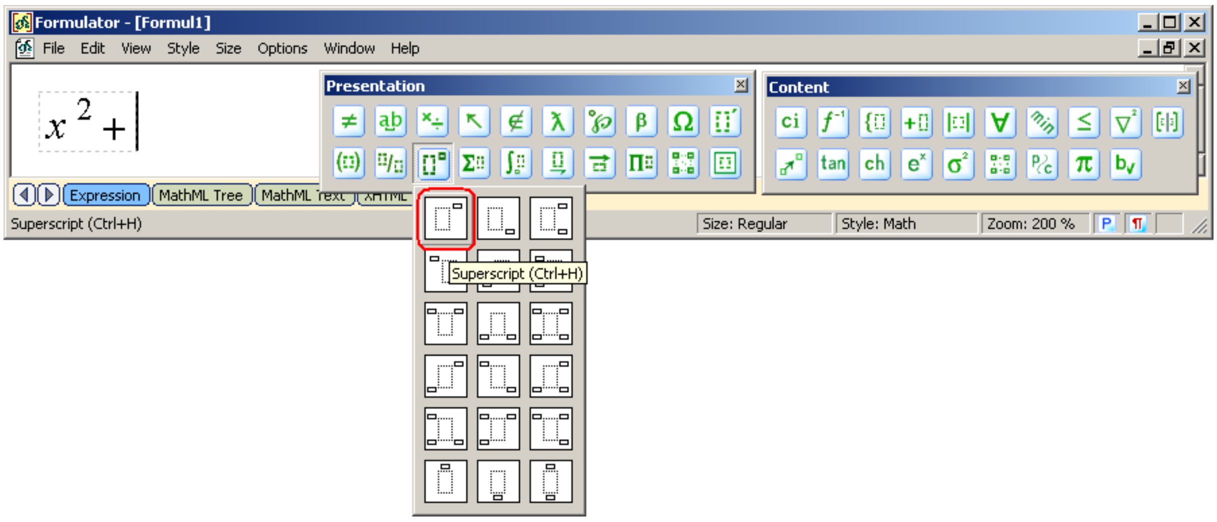

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Presentation
Content
Expression MathML Tree MathML Text XHTML
Size: Subscript Style: Math Zoom: 200 %

Navigation: line - content MathML; parent node - "apply" ("power") element

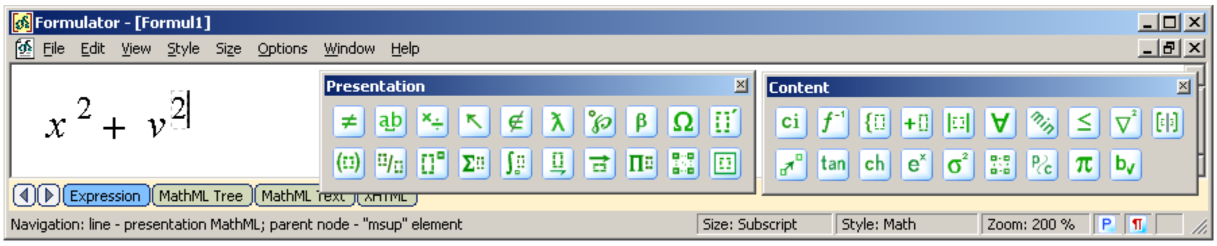
Switch to the Presentation MathML input mode; type ‘+’.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Presentation
Content
Expression MathML Tree MathML Text XHTML
Size: Regular Style: Math Zoom: 200 %

Press the Right arrow to get out of the “apply” element; using presentation toolbar insert the superscript MathML node.

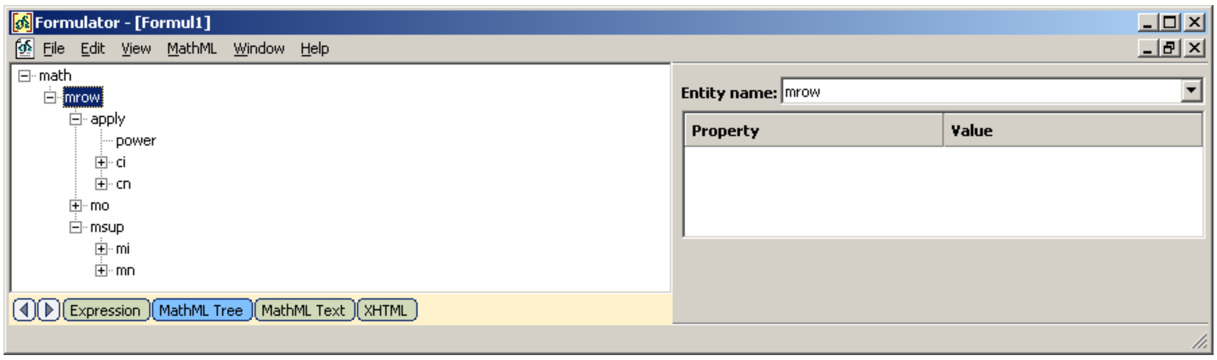


Press ‘v’, the Right arrow, ‘2’.

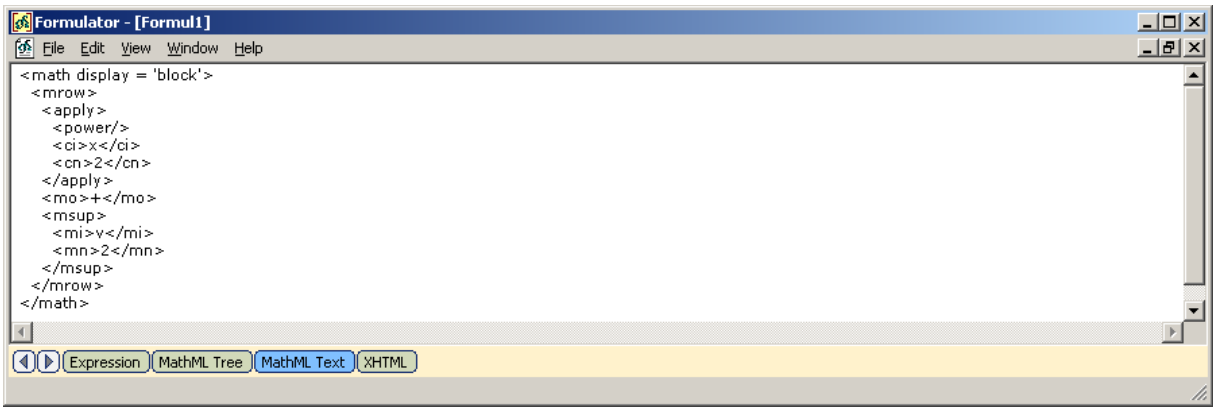

Formulator - [Formul1]
File Edit View Style Size Options Window Help
Presentation
Content
Expression MathML Tree MathML Text XHTML
Size: Subscript
Style: Math
Zoom: 200 %

Navigation: line - presentation MathML; parent node - "msup" element

See the resulting MathML tree and text.


Formulator - [Formul1]
File Edit View MathML Window Help
Entity name: mrow
Property
Value
Expression MathML Tree MathML Text XHTML

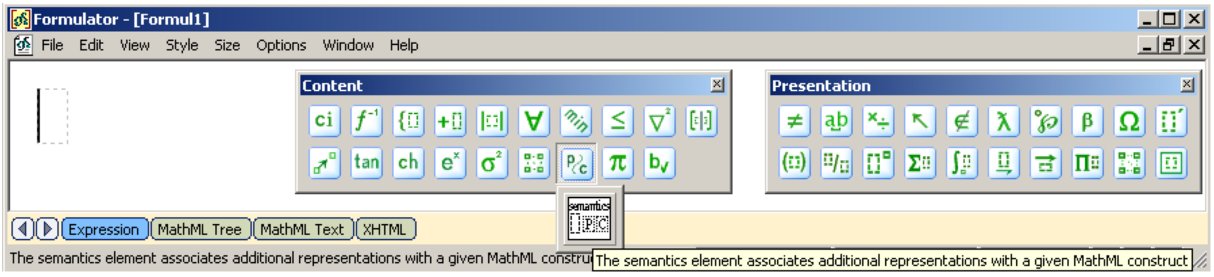
```
math
  mrow
    apply
      power
      ci
      cn
    mo
    msup
      mi
      mn
```


Formulator - [Formul1]
File Edit View Window Help
Expression MathML Tree MathML Text XHTML

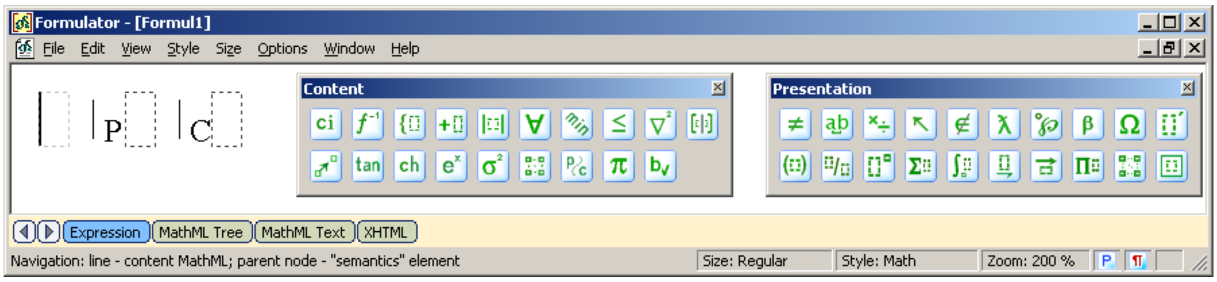
```
<math display = 'block'>
 <mrow>
  <apply>
   <power/>
   <ci>x</ci>
   <cn>2</cn>
  </apply>
  <mo>+</mo>
  <msup>
   <mi>v</mi>
   <mn>2</mn>
  </msup>
 </mrow>
</math>
```

3. By using Semantic Mapping Elements from the toolbar content markup can be contained in presentation markup as a *Parallel Markup*, thus making use of both presentation and content information for the same element.

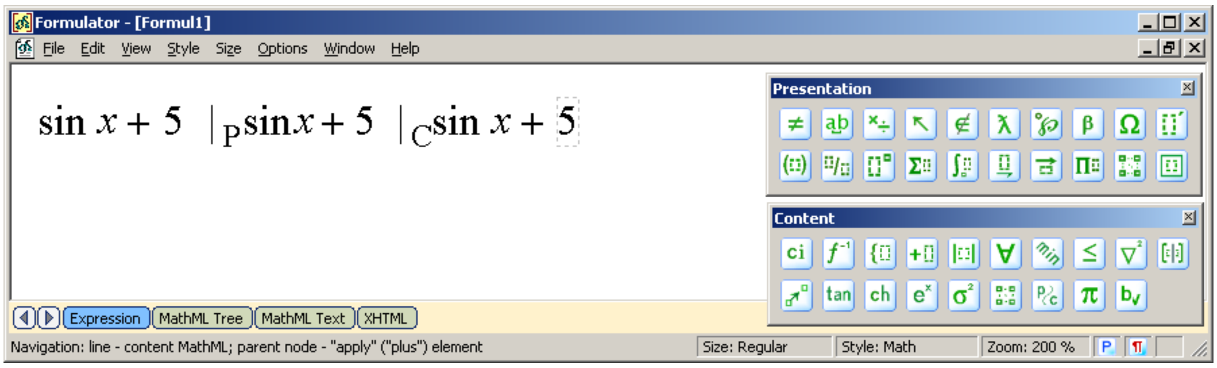
Insert the "semantics" element.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Presentation
semantics
Expression MathML Tree MathML Text XHTML
The semantics element associates additional representations with a given MathML construct

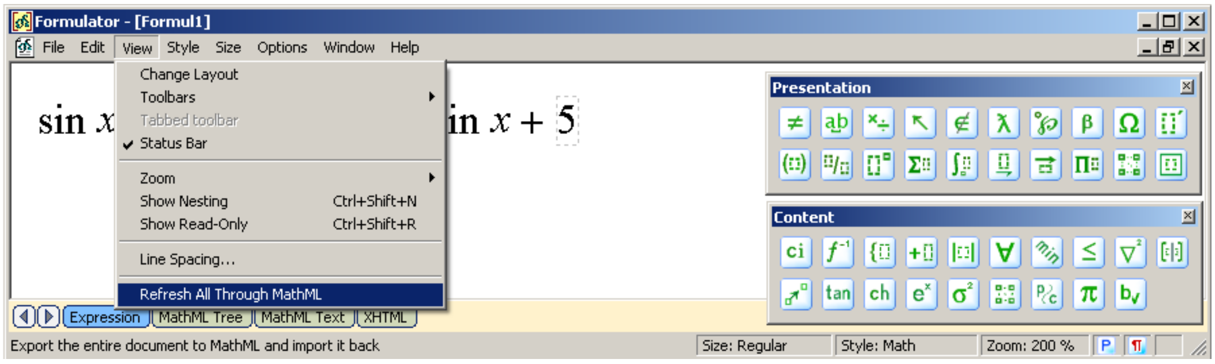
Its rendering can't be correct right away, since a user should input needed values of child "annotation" elements.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Content
Presentation
Expression MathML Tree MathML Text XHTML
Navigation: line - content MathML; parent node - "semantics" element
Size: Regular Style: Math Zoom: 200 %

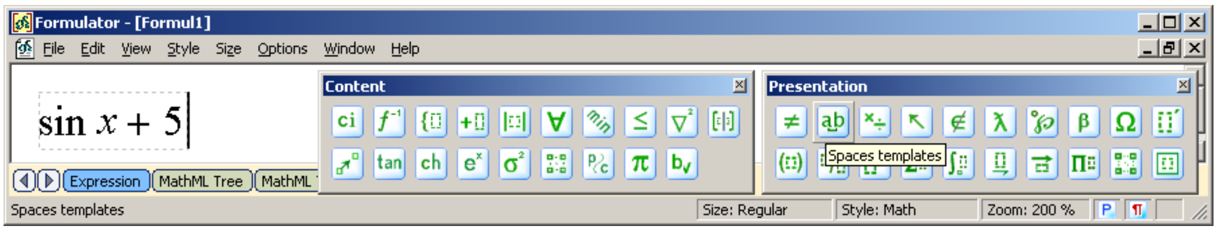
Create "sin(x) + 5" expression using different markups.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Presentation
Content
Expression MathML Tree MathML Text XHTML
Navigation: line - content MathML; parent node - "apply" ("plus") element
Size: Regular Style: Math Zoom: 200 %

Refresh appearance of the document.


Formulator - [Formul1]
File Edit View Style Size Options Window Help
Change Layout
Toolbars
Tabbed toolbar
Status Bar
Zoom
Show Nesting Ctrl+Shift+N
Show Read-Only Ctrl+Shift+R
Line Spacing...
Refresh All Through MathML
Presentation
Content
Expression MathML Tree MathML Text XHTML
Export the entire document to MathML and import it back
Size: Regular Style: Math Zoom: 200 %

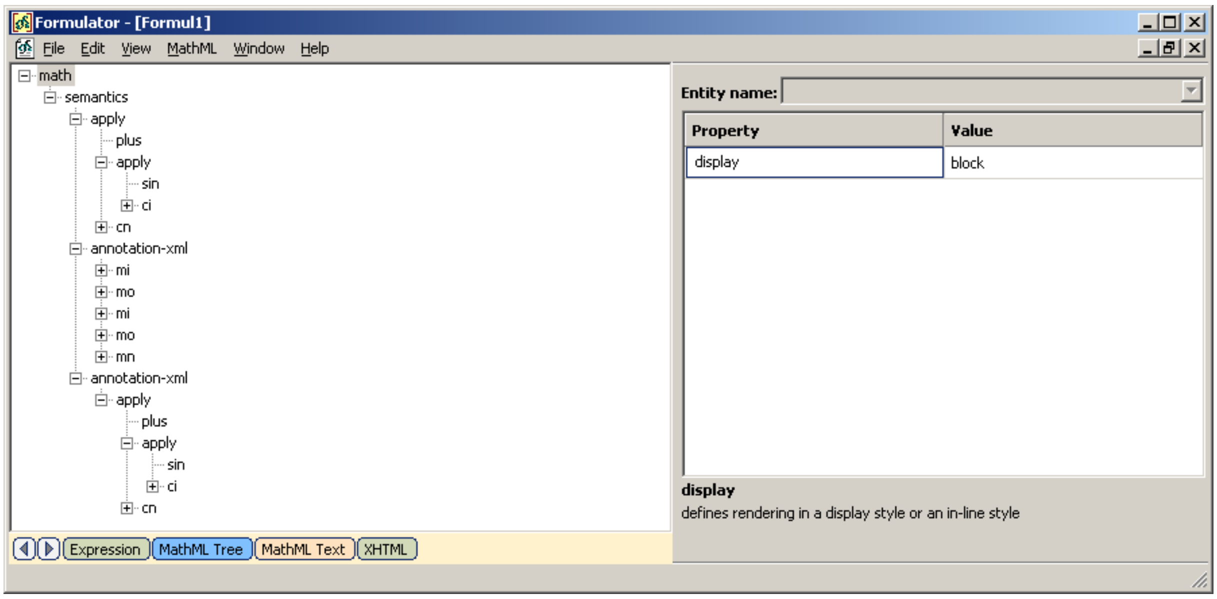
See results on different pages of MathML Weaver.



$\sin x + 5$

Spaces templates

Size: Regular | Style: Math | Zoom: 200 %

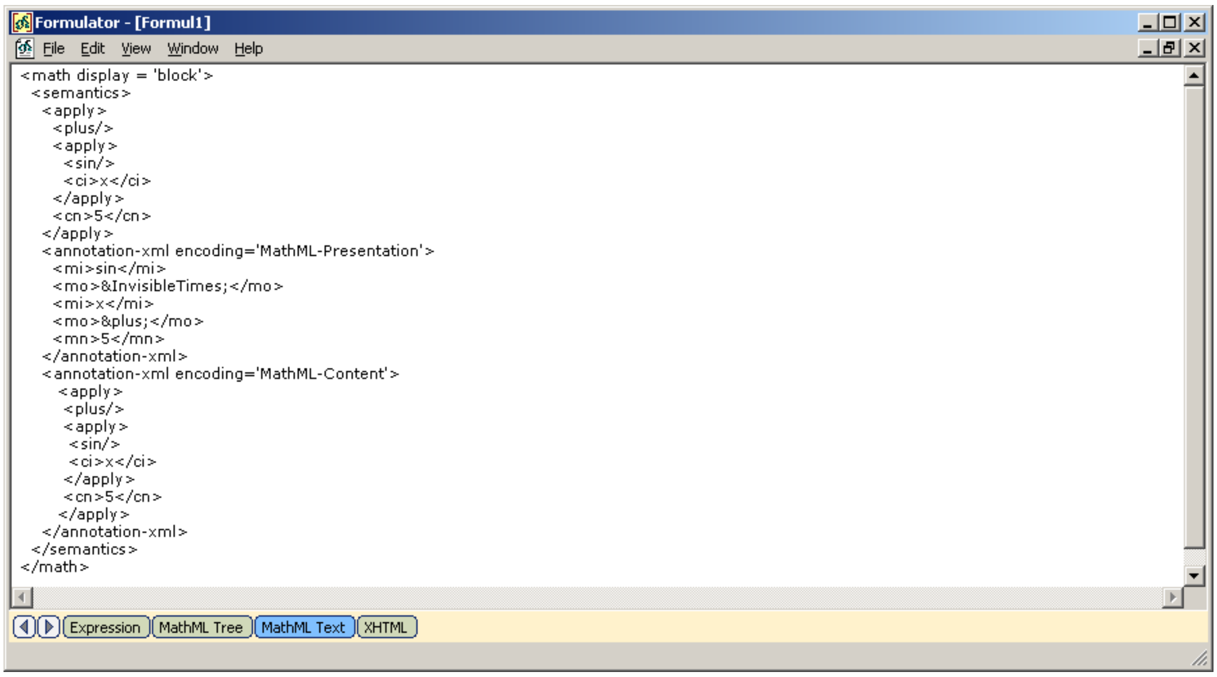


```
math
  semantics
    apply
      plus
      apply
        sin
        ci
      cn
    annotation-xml
      mi
      mo
      mi
      mo
      mn
    annotation-xml
      apply
        plus
        apply
          sin
          ci
        cn
```

Entity name:

Property	Value
display	block

display
defines rendering in a display style or an in-line style



```
<math display = 'block'>
 <semantics>
  <apply>
   <plus/>
   <apply>
    <sin/>
    <ci>x</ci>
   </apply>
   <cn>5</cn>
  </apply>
  <annotation-xml encoding='MathML-Presentation'>
   <mi>sin</mi>
   <mo>&InvisibleTimes;</mo>
   <mi>x</mi>
   <mo>&plus;</mo>
   <mn>5</mn>
  </annotation-xml>
  <annotation-xml encoding='MathML-Content'>
    <apply>
     <plus/>
     <apply>
      <sin/>
      <ci>x</ci>
     </apply>
     <cn>5</cn>
    </apply>
  </annotation-xml>
 </semantics>
</math>
```