Hermitech Laboratory

Address: *Chapaeva str., 7, Zhytomyr, 10029, Ukraine*
Phone: *+38-0412-447214*
Fax: *+38-0412-480198*
Web: *http://www.mmlsoft.com*
E-mail: *info@mmlsoft.com*

FORMULATOR
v3.8

ActiveX Control

Developer Manual

Published By
Hermitech Laboratory
7, Chapaeva str.,
Zhytomyr, 10029
Ukraine

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

Table of Contents

INTRODUCTION 6
HOW TO REDISTRIBUTE 7
Installing parameters 7
Uninstalling parameters 8
FORMULATOR ACTIVEX CONTROL PROPERTIES 10
BackgroundColor 12
EnableMathShortcuts 12
FontSize 19
ToolBar 19
StatusBar 19
IsModified 19
ShowNesting 20
ShowReadOnly 20
LineSpacing 20
Scale 20
MathMLNamespace 20
TopIndent 21
BottomIndent 21
LeftIndent 22
RightIndent 22
DefaultView 22
TabbedLayout 22
ExpressionBar 22
MathMLTreeBar 22
MathMLTreePropertyBar 22
MathMLTextBar 23
XHTMLBar 23
MathMLMath 23
SymbolName4Unicode 23
DisplayTimes 23
ShowInvisibleElements 23
ShowNavigationInfo 23
ShowRefreshMahML 24
MultiViewBar 24
ContextInputMode 24
CurrentView 24
EnableContextMenu 24
SimplifiedContextMenu 25
FORMULATOR ACTIVEX CONTROL METHODS 26
Copy 28
Cut 28
Paste 28
Undo 29
Redo 29

SelectAll 29
DialogOtherStyle 29
DialogDefineStyle 29
DialogOtherSize 29
DialogDefineSize 30
SetStyleFaceName 30
GetStyleFaceName 30
SetStyleBold 31
GetStyleBold 31
SetStyleItalic 31
GetStyleItalic 31
SetStyleColor 31
GetStyleColor 31
SetExpressionColor 31
SetSymbolSize 31
GetSymbolSize 31
SetStyle 32
GetStyle 33
SetSizeLargeOp 33
SetSizeRegular 33
SetSizeSubscript 33
SetSizeSubSubscript 33
SetSizeSmaller 34
SetSizeLarger 34
SetMathML 34
GetMathML 34
GetMathMLStr 34
ExportEMF 34
ExportHEMF 34
ExportImage 35
ExportHBitmap 35
ExportXHTML 35
DrawExpression 35
RelayMessage 35
GetFormulatorLicenseInfo 36
EmulateToolbarButton 36
SetRichUIMode 62
IsRichUIMode 62
OnChange 62
OnEnter 62
OnChar 63
OnSetFocus 63
OnKillFocus 63
OnLMouseClick 63
OnLMouseDblClick 63
OnRMouseClick 63

OnRMouseDblClick 63

INTRODUCTION

Formulator is a dynamic and intelligent mathematical equation editor designed for personal computers running Microsoft® Windows. This application allows you to create mathematical equations through simple point-and-click techniques. Equations can be converted into MathML, or the other textual languages using a customizable translation mechanism, and can be saved in several graphic file formats, ready to be imported into documents. Formulator supports not only import into the MathML format, but also export from this approved standard for math on the World Wide Web.

For each basic mathematical construct, Formulator provides a template containing graphics and edit boxes. There are several template groups, including fractions, radicals, sums, matrices, various types of brackets and braces, etc. Users create equations by inserting templates and filling in their edit boxes. Users can insert templates into the edit boxes of other templates, and in that way complex hierarchical formulas can be built up.

Formulator comprises both a standalone application edition, that can be run as a separate program, and a component edition, that is, an Active Document Server or an ActiveX control, which can be incorporated into any Active Document Container or ActiveX Container accordingly.

The editing functionality provided by Formulator, when combined with the API of the Formulator component edition, can be used to implement a variety of mathematical applications that can be utilized in several different contexts. For example, an interface for computer algebra systems can be created that interchange MathML data, or multimedia materials with mathematical data can be created.

HOW TO REDISTRIBUTE

To distribute your application using Formulator ActiveX Control, you must redistribute end-user version of the Formulator ActiveX Control to the client machines as part of your application. This is accomplished by installing the redistribution pack of the Formulator ActiveX Control (*fmlsetup-ax-redist.exe*) along with your application. This redistribution pack includes only binaries which are mandatory for running of the Formulator ActiveX Control (without help file and user manuals).

Installing parameters

Note that you are prohibited from distributing the developer license file (fmlaxc.lic) according to the license agreement. Please refer to the license agreement to resolve other licensing questions.

The redistribution pack of the Formulator ActiveX Control accepts optional command line parameters. They are listed below, according to the Inno Setup documentation.

/SILENT, /VERYSILENT

Instructs Setup to be silent or very silent.

When Setup is silent the wizard and the background window are not displayed but the installation progress window is.

When a setup is very silent this installation progress window is not displayed. Everything else is normal so for example error messages during installation are displayed.

/SUPPRESSMSGBOXES

Instructs Setup to suppress message boxes. Only has an effect when combined with '/SILENT' and '/VERYSILENT'.

The default response in situations where there's a choice is:

- Yes in a 'Keep newer file?' situation.
- No in a 'File exists, confirm overwrite.' situation.
- Abort in Abort/Retry situations.
- Cancel in Retry/Cancel situations.
- Yes (=continue) in a DiskSpaceWarning/ DirExists/ DirDoesntExist/ NoUninstallWarning/ ExitSetupMessage/ ConfirmUninstall situation.
- Yes (=restart) in a FinishedRestartMessage/ UninstalledAndNeedsRestart situation.

Five message boxes are not suppressible:

- The About Setup message box.
- The Exit Setup? message box.
- The FileNotInDir2 message box displayed when Setup requires a new disk to be inserted and the disk was not found.

- Any (error) message box displayed before Setup (or Uninstall) could read the command line parameters.
- Any message box displayed by [Code] support function MsgBox.

/DIR="x:\dirname"

Overrides the default directory name displayed on the Select Destination Location wizard page. A fully qualified pathname must be specified.

/LOG

Causes Setup to create a log file in the user's TEMP directory detailing file installation and [Run] actions taken during the installation process.

This can be a helpful debugging aid. For example, if you suspect a file isn't being replaced when you believe it should be (or vice versa), the log file will tell you if the file was really skipped, and why.

The log file is created with a unique name based on the current date. (It will not overwrite or append to existing files.)

The information contained in the log file is technical in nature and therefore not intended to be understandable by end users. Nor is it designed to be machine-parseable; the format of the file is subject to change without notice.

/LOG="filename"

Same as /LOG, except it allows you to specify a fixed path/filename to use for the log file. If a file with the specified name already exists it will be overwritten. If the file cannot be created, Setup will abort with an error message.

Uninstalling parameters

After installation of the Formulator ActiveX Control, its folder contains uninstaller program 'unins000.exe'. This uninstaller program may be used during your application uninstalling in order to remove Formulator ActiveX Control. The uninstaller program accepts the following optional command line parameters (according to the Inno Setup documentation).

/SILENT, /VERYSILENT

When specified, the uninstaller will not ask the user for startup confirmation or display a message stating that uninstall is complete. Shared files that are no longer in use are deleted automatically without prompting. Any critical error messages will still be shown on the screen. When '/VERYSILENT' is specified, the uninstallation progress window is not displayed.

If a restart is necessary and the '/NORESTART' command isn't used (see below) and '/VERYSILENT' is specified, the uninstaller will reboot without asking.

/SUPPRESSMSGBOXES

Instructs the uninstaller to suppress message boxes. Only has an effect when combined with '/SILENT' and '/VERYSILENT'. See '/SUPPRESSMSGBOXES' under redistribution pack command line parameters for more details.

/LOG

Causes Uninstall to create a log file in the user's TEMP directory detailing file uninstallation and [UninstallRun] actions taken during the uninstallation process. This can be a helpful debugging aid.

The log file is created with a unique name based on the current date. (It will not overwrite or append to existing files.)

The information contained in the log file is technical in nature and therefore not intended to be understandable by end users. Nor is it designed to be machine-parseable; the format of the file is subject to change without notice.

/NORESTART

Instructs the uninstaller not to reboot even if it's necessary.

FORMULATOR ACTIVEX CONTROL PROPERTIES

The properties of the Formulator ActiveX Control are listed below.

Properties editing in the free version of the component edition (Formulator Express ActiveX Control) is restricted: all properties are accessible for reading and only four properties can be set (*ToolBar*, *StatusBar*, *TabbedLayout*, *ExpressionBar*).

Property Name	Property Description
BackgroundColor	Sets the background color of the ActiveX Control.
EnableMathShortcuts	If set to *true* all of the default keyboard shortcuts for the mathematical buttons insertion will be active. Setting this property to *false* deactivates default keyboard shortcuts for the mathematical buttons insertion and allows you to define your own user defined keyboard shortcuts.
FontSize	Gets or sets the current font size in points (pt).
ToolBar	This property allows you show and hide the built-in toolbar of the Formulator ActiveX Control's.
StatusBar	This property allows you show and hide the built-in statusbar of the Formulator ActiveX Control's.
IsModified	This property allows you to determine whether a document was changed.
ShowNesting	This property toggles the equation display between normal viewing mode and nesting mode.
ShowReadOnly	This property toggles the equation display between normal viewing mode and "show read-only nodes" mode.
LineSpacing	This property changes default value of distance between neighbour lines (in %).
Scale	This property allows you to change the viewing scale (in %). Acceptable values of the viewing scale are: 100, 200, 300, 500, 1000.
MathMLNamespace	This item selects whether a namespace will be used while converting mathematical expression to MathML (no namespace, *m* namespace, namespace attribute).
TopIndent	Gets or sets the top indent for a document in pixels (px).
BottomIndent	Gets or sets the bottom indent for a document in pixels (px).
LeftIndent	Gets or sets the left indent for a document in

	pixels (px).
RightIndent	Gets or sets the right indent for a document in pixels (px).
DefaultView	Gets or sets the default editing option (0 - Expression, 1 - MathML Tree, 2 - MathML Text).
TabbedLayout	Gets or sets the default layout (true - one tabbed toolbar, false - normal docked toolbars).
ExpressionBar	Shows or hides the Standard toolbar for the "Expression" editing option.
MathMLTreeBar	Shows or hides the Standard toolbar for the "MathML Tree" editing option.
MathMLTreePropertyBar	Shows or hides the Property pane for the "MathML Tree" editing option.
MathMLTextBar	Shows or hides the Standard toolbar for the "MathML Text" editing option.
XHTMLBar	Shows or hides the Standard toolbar for the "XHTML" editing option.
MathMLMath	Gets or sets the option how to render the <math> element in XHTML (0 — 'block', 1 — 'inline' mode).
SymbolName4Unicode	Gets or sets the option that defines whether Formulator should try to find the equivalent entity name for Unicode characters while converting expressions into MathML (otherwise the corresponding number will be used).
DisplayTimes	Gets or sets the option how to render the <times/> element in the Content markup. It can be rendered using such presentation elements as: 0 — "&times;", 1 — "&sdot;", and 2 — "&InvisibleTimes;".
ShowInvisibleElements	Gets or sets the option whether to render invisible Content MathML elements.
ShowNavigationInfo	Gets or sets the option whether Formulator should display navigation information about the current and the parent node in the Status Bar.
ShowRefreshMahML	Gets or sets the option whether Formulator should display notification to a user when the current document needs to be refreshed through MathML. This feature is needed when changing of an option influences all or part of text presentation. E.g., it could be the case of using another symbol for <times/> element of the Content markup, or beautifying of such a formula in Content markup that uses additional slots for inputting <bvar> elements.

MultiViewBar	Hides or shows the list of all existing editing modes.
ContextInputMode	Switches between Content and Presentation MathML Input Mode (the last one is used by default). 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 Presentation MathML Input Mode leads to inserting of the <mo> element (Presentation markup); in the Content MathML Input Mode such an action inserts a mathematical template (?+?) for the <apply> element with the operator element <plus/>.
CurrentView	Gets or sets the current editing option (0 – Expression, 1 – MathML Tree, 2 – MathML Text, 3 – XHTML).
EnableContextMenu	Control if context menu is enabled
SimplifiedContextMenu	Switches a context menu to the simplified form (edit commands Cut, Copy, Paste, etc.)

BackgroundColor

Sets the background color of the ActiveX Control.

public long BackgroundColor {get; set;}

EnableMathShortcuts

If set to *true* all of the default keyboard shortcuts for the mathematical buttons insertion will be active. Setting this property to *false* deactivates default keyboard shortcuts for the mathematical buttons insertion and allows you to define your own user defined keyboard shortcuts.

public bool EnableMathShortcuts {get; set;}

You can execute some Formulator operations directly from the keyboard via shortcuts. The full list of the default keyboard shortcuts for the mathematical buttons insertion is stated in the table below. Note that some shortcuts require you to type two keystroke combinations consecutively.

Mathematical template	**Keyboard shortcut**
Commands for items of the "Relational and logical symbols" toolbar:	
Less-than or equal to	Ctrl+K,,
Greater-than or equal to	Ctrl+K,.
Not equal to	Ctrl+K,+
Identical to	Ctrl+K,=
Tilde operator	Ctrl+K,Alt+~
Almost equal to	Ctrl+K,~
Proportional to	Ctrl+K,P
Therefore	Ctrl+Shift+K,T
There exists	Ctrl+Shift+K,E
For all	Ctrl+Shift+K,A
Not sign	Ctrl+Shift+K,N
Logical and	Ctrl+Shift+K,7
Logical or	Ctrl+Shift+K,\
Commands for items of the "Spaces templates" toolbar:	
1-point space	Ctrl+Alt+Space
Thin space (1/6 EM)	Ctrl+Space
Thick space (1/3 EM)	Ctrl+Shift+Space
EM space (Ctrl+K,4	
Commands for items of the "Operator symbols" toolbar:	
Plus-minus sign	Ctrl+Shift+K,=
Multiplication sign	Ctrl+K,T
Asterik operator	Ctrl+Shift+K,*
Division sign	Ctrl+Shift+K,/
Dot operator	Ctrl+Shift+K,.
Bullet	Ctrl+Shift+K,8
Left-pointing angle bracket	Ctrl+Shift+K,<
Right-pointing angle bracket	Ctrl+Shift+K,>
Commands for items of the "Arrow symbols" toolbar:	
Left right arrow	Ctrl+K,Alt+Left
Rightwards arrow	Ctrl+K,Right
Leftwards arrow	Ctrl+K,Left
Up down arrow	Ctrl+K,Alt+Up
Upwards arrow	Ctrl+K,Up
Downwards arrow	Ctrl+K,Down
Left right double arrow	Ctrl+K,Alt+Shift+Left
Rightwards double arrow	Ctrl+K,Shift+Right
Leftwards double arrow	Ctrl+K,Shift+Left

Up down double arrow	Ctrl+K,Alt+Shift+Up
Upwards double arrow	Ctrl+K,Shift+Up
Downwards double arrow	Ctrl+K,Shift+Down
Rightwards arrow from bar	Ctrl+K,Tab
Downwards arrow with corner leftwards	Ctrl+K,Enter
Commands for items of the "Set theory symbols" toolbar:	
Element of	Ctrl+K,E
Not an element of	Ctrl+K,Shift+E
Union	Ctrl+K,U
Intersection	Ctrl+K,X
Subset	Ctrl+K,C
Superset	Ctrl+K,S
Not a subset of	Ctrl+K,Shift+C
Empty set	Ctrl+K,O
Commands for items of the "Special constants" toolbar:	
Partial Differential	Ctrl+K,D
Greek small letter pi	Ctrl+G,P
Planck constant over two pi	Ctrl+K,H
Infinity	Ctrl+K,I
Latin small letter lambda with stroke	Ctrl+K,L
Script small l	Ctrl+Shift+K,L
Commands for items of the "Miscellaneous symbols" toolbar:	
Fraktur capital I, imaginary part	Ctrl+K,Shift-I
Fraktur capital R, real part	Ctrl+K,Shift-R
Alef symbol	Ctrl+K,A
Blackboard-bold capital R, the set of all real numbers	Ctrl+D,Shift+R
Blackboard-bold capital Z, the set of all integer numbers	Ctrl+D,Shift+Z
Blackboard-bold capital C, the set of all rational numbers	Ctrl+D,Shift+C
Blackboard-bold capital Q, the set of all rational numbers	Ctrl+D,Shift+Q
Blackboard-bold capital N, the set of all natural numbers	Ctrl+D,Shift+N
Greek capital letter delta	Ctrl+G,Shift+D
Greek capital letter omega	Ctrl+G,Shift+W
Inverted ohm sign	Ctrl+Shift+K,Shift+O
Degree sign	Ctrl+Shift+K,D
Angle	Ctrl+Shift+K,Shift+A

Measured angle	Ctrl+Shift+K,Alt+A
Spherical angle	Ctrl+Shift+K,Alt+Shift+A
Perpendicular	Ctrl+Shift+K,P
Parallel	Ctrl+Shift+K,I
Commands for items of the "Greek characters (lowercase)" toolbar:	
Greek small letter alpha	Ctrl+G,A
Greek small letter beta	Ctrl+G,B
Greek small letter gamma	Ctrl+G,G
Greek small letter delta	Ctrl+G,D
Greek small letter epsilon	Ctrl+G,E
Greek small letter zeta	Ctrl+G,Z
Greek small letter eta	Ctrl+G,H
Greek small letter theta	Ctrl+G,Q
Greek small letter iota	Ctrl+G,I
Greek small letter kappa	Ctrl+G,K
Greek small letter lambda	Ctrl+G,L
Greek small letter mu	Ctrl+G,M
Greek small letter nu	Ctrl+G,N
Greek small letter xi	Ctrl+G,X
Greek small letter omicron	Ctrl+G,O
Greek small letter pi	Ctrl+G,P
Greek small letter rho	Ctrl+G,R
Greek sigma symbol	Ctrl+G,Shift+V
Greek small letter sigma	Ctrl+G,S
Greek small letter tau	Ctrl+G,T
Greek small letter upsilon	Ctrl+G,U
Greek small letter phi	Ctrl+G,F
Greek phi symbol	Ctrl+G,J
Greek small letter chi	Ctrl+G,C
Greek small letter psi	Ctrl+G,Y
Greek small letter omega	Ctrl+G,W
Commands for items of the "Greek characters (uppercase)" toolbar:	
Greek capital letter alpha	Ctrl+G,Shift+A
Greek capital letter beta	Ctrl+G,Shift+B
Greek capital letter gamma	Ctrl+G,Shift+G
Greek capital letter delta	Ctrl+G,Shift+D
Greek capital letter epsilon	Ctrl+G,Shift+E
Greek capital letter zeta	Ctrl+G,Shift+Z
Greek capital letter eta	Ctrl+G,Shift+H
Greek capital letter theta	Ctrl+G,Shift+Q
Greek capital letter iota	Ctrl+G,Shift+I

Greek capital letter kappa	Ctrl+G,Shift+K
Greek capital letter lambda	Ctrl+G,Shift+L
Greek capital letter mu	Ctrl+G,Shift+M
Greek capital letter nu	Ctrl+G,Shift+N
Greek capital letter xi	Ctrl+G,Shift+X
Greek capital letter omicron	Ctrl+G,Shift+O
Greek capital letter pi	Ctrl+G,Shift+P
Greek capital letter rho	Ctrl+G,Shift+R
Greek capital letter sigma	Ctrl+G,Shift+S
Greek capital letter tau	Ctrl+G,Shift+T
Greek capital letter phi	Ctrl+G,Shift+F
Greek capital letter chi	Ctrl+G,Shift+C
Greek capital letter psi	Ctrl+G,Shift+Y
Greek capital letter omega	Ctrl+G,Shift+W
Commands for items of the "Differentiation templates" toolbar:	
Prime	Ctrl+Alt+’
Double prime	Ctrl+"
Commands for items of the "Fence templates" toolbar:	
Parenthesis or round-brackets	Ctrl+9
Brackets or square-brackets	Ctrl+[
Braces or curly-brackets	Ctrl+{
Angle brackets	Ctrl+,
Single vertical bars	Ctrl+T, |
Left brackets	Ctrl+T,[
Right brackets	Ctrl+T,]
Left braces	Ctrl+T,{
Right braces	Ctrl+T,}
Left angle brackets	Ctrl+T,<
Right angle brackets	Ctrl+T,>
Commands for items of the "Fraction and radical templates" toolbar:	
Full-size fraction	Ctrl+F
Full-size diagonal fraction	Ctrl+/
Slash fraction	Ctrl+T, Alt+/
Square root	Ctrl+R
nth root	Ctrl+T,N
Commands for items of the "Subscript and superscript templates" toolbar:	
Superscript	Ctrl+H
Subscript	Ctrl+L
Superscript and subscript	Ctrl+J

Over-script	Ctrl+T, Shift+L
Under-script (limit)	Ctrl+T, Alt-L
Over-script and under-script	Ctrl+T, L
Commands for items of the "Summation templates" toolbar:	
Summation with no limits	Ctrl+T,Shift+S
Summation with underscript limit	Ctrl+T,Alt+S
Summation with underscript and overscript	Ctrl+T,S
Commands for items of the "Integral templates" toolbar:	
Indefinite integral (no limits)	Ctrl+Shift+I,!
Definite integral with underscript and overscript limits	Ctrl+Shift+I,S
Definite integral with subscript and superscript limit	Ctrl+I
Definite integral with underscript limit	Ctrl+Shift+I,Alt+S
Definite integral with subscript limit	Ctrl+Shift+I,Alt+1
Double integral with no limits	Ctrl+Shift+I,@
Double integral with underscript limit	Ctrl+Shift+I,2
Double integral with subscript limit	Ctrl+Shift+I,Alt+2
Triple integral with no limits	Ctrl+Shift+I,#
Triple integral with underscript limit	Ctrl+Shift+I,3
Triple integral with subscript limit	Ctrl+Shift+I,Alt+3
Contour integral with no limits	Ctrl+Shift+I,Shift+C
Contour integral with underscript limit	Ctrl+Shift+I,C
Contour integral with subscript limit	Ctrl+Shift+I,Alt+C
Area integral with no limits	Ctrl+Shift+I,Shift+A
Area integral with underscript limit	Ctrl+Shift+I,A
Area integral with subscript limit	Ctrl+Shift+I,Alt+A
Volume integral with no limits	Ctrl+Shift+I,Shift+V
Volume integral with underscript limit	Ctrl+Shift+I,V
Volume integral with subscript limit	Ctrl+Shift+I,Alt+V
Integral with counter-clockwise loop with no limits	Ctrl+Shift+I,Shift+Left
Integral with counter-clockwise loop with underscript limit	Ctrl+Shift+I,Left
Integral with counter-clockwise loop with subscript limit	Ctrl+Shift+I,Alt+Left
Integral with clockwise loop with no limits	Ctrl+Shift+I,Shift+Right
Integral with clockwise loop with underscript limit	Ctrl+Shift+I,Right
Integral with clockwise loop with	Ctrl+Shift+I,Alt+Right

subscript limit	
Commands for items of the "Underbar and overbar templates" toolbar:	
Tilde	Ctrl+^,~
Hat	Ctrl+^,6
Arc	Ctrl+^,9
Joint status	Ctrl+^,J
Over-bar	Ctrl+^,-
Double over-bar	Ctrl+^,D
Under-bar	Ctrl+^,_
Double Under-bar	Ctrl+^,Shift+D
Right arrow over-bar	Ctrl+^,Right
Left arrow over-bar	Ctrl+^,Left
Right harpoon over-bar	Ctrl+^,Alt+Right
Double-headed arrow over-bar	Ctrl+^,Up
Right arrow under-bar	Ctrl+^,Shift+Right
Left arrow under-bar	Ctrl+^,Shift+Left
Right harpoon under-bar	Ctrl+^,Alt+Shift+Right
Double-headed arrow under-bar	Ctrl+^,Shift+Up
Mid-line strike through	Ctrl+^,Alt+-
Strike through	Ctrl+^,X
Strike-through (bottom-left to upper-right)	Ctrl+^,/
Strike-through (top-left to bottom-right)	Ctrl+^,\
Commands for items of the "Labeled arrow templates" toolbar:	
Right arrow with upper text slot	Ctrl+T,Shift+Right
Right arrow with lower text slot	Ctrl+T,Alt+Right
Right arrow with upper and lower text slot	Ctrl+T,Right
Left arrow with upper text slot	Ctrl+T,Shift+Left
Left arrow with lower text slot	Ctrl+T,Alt+Left
Left arrow with upper and lower text slot	Ctrl+T,Left
Double-headed arrow with upper text slot	Ctrl+T,Shift+Up
Double-headed arrow with lower text slot	Ctrl+T,Alt+Up
Double-headed arrow with upper and lower text slot	Ctrl+T,Up
Commands for items of the "Products and set theory templates" toolbar:	
Product with no limits	Ctrl+T,Shift+P
Product with underscript limit	Ctrl+T,Alt+P

Product with underscript and overscript limits	Ctrl+T,P
Coproduct with no limits	Ctrl+T,Shift+C
Coproduct with underscript limit	Ctrl+T,Alt+C
Coproduct with underscript and overscript limits	Ctrl+T,C
Intersection with no limits	Ctrl+T,Shift+I
Intersection with underscript limit	Ctrl+T,Alt+I
Intersection with underscript and overscript limits	Ctrl+T,I
Union with no limits	Ctrl+T,Shift+U
Union with underscript limit	Ctrl+T,Alt+U
Union with underscript and overscript limits	Ctrl+T,U
Commands for items of the "Box templates" toolbar:	
Box	Ctrl+Shift+T,X

FontSize

Gets or sets the current font size in points (pt).

public long FontSize {get; set;}

ToolBar

This property allows you show and hide the built-in toolbar of the Formulator ActiveX Control's.

public bool ToolBar {get; set;}

StatusBar

This property allows you show and hide the built-in statusbar of the Formulator ActiveX Control's.

public bool StatusBar {get; set;}

IsModified

This property allows you to determine whether a document was changed.

public bool IsModified{get; set;}

ShowNesting

This property toggles the equation display between normal viewing mode and nesting mode where the you can see the hierarchical structure of your equations.

```
public bool ShowNesting{get; set;}
```

ShowReadOnly

This property toggles the equation display between normal viewing mode and "show read-only nodes" mode where the you can see highlighted that nodes which can't be changed.

```
public bool ShowReadOnly{get; set;}
```

LineSpacing

This property changes default value of distance between neighbour lines (in %).

```
public long LineSpacing{get; set;}
```

Scale

This property allows you to change the viewing scale (in %). Acceptable values of the viewing scale are: 100, 200, 300, 500, 1000.

```
public long Scale{get; set;}
```

MathMLNamespace

This item selects whether a namespace will be used while converting mathematical expression to MathML. Acceptable values of this property are *long*:

```
public long MathMLNamespace{get; set;}

#define MMLNS_NO_NAMESPACE      0
#define MMLNS_M                 1
#define MMLNS_ATTR              2
```

- *MMLNS_NO_NAMESPACE* (no namespace)

$\frac{1}{2}$ ⇒

```
<math display = ‘block’>
 <mrow>
  <mfrac>
   <mn>1</mn>
   <mn>2</mn>
  </mfrac>
 </mrow>
</math>
```

- *MMLNS_M* (m namespace)

$\frac{1}{2}$ ⇒

```
<m:math display = ‘block’>
 <m:mrow>
  <m:mfrac>
   <m:mn>1</m:mn>
   <m:mn>2</m:mn>
  </m:mfrac>
 </m:mrow>
</m:math>
```

- *MMLNS_ATTR* (namespace attribute)

$\frac{1}{2}$ ⇒

```
<math   display = ‘block’
        xmlns=‘http://www.w3.org/1998/Math/MathML’>
 <mrow>
  <mfrac>
   <mn>1</mn>
   <mn>2</mn>
  </mfrac>
 </mrow>
</math>
```

TopIndent

Gets or sets the top indent for a document in pixels (px).

public long TopIndent {get; set;}

BottomIndent

Gets or sets the bottom indent for a document in pixels (px).

public long BottomIndent {get; set;}

LeftIndent

Gets or sets the left indent for a document in pixels (px).

public long LeftIndent {get; set;}

RightIndent

Gets or sets the right indent for a document in pixels (px).

public long RightIndent{get; set;}

DefaultView

Gets or sets the default editing option (0 — "Expression", 1 — "MathML Tree", 2 — "MathML Text", 3 — "XHTML").

public long DefaultView {get; set;}

TabbedLayout

Gets or sets the default layout (true — one tabbed toolbar, false — normal docked toolbars).

public bool TabbedLayout {get; set;}

ExpressionBar

Shows or hides the Standard toolbar for the "Expression" editing option.

public bool ExpressionBar{get; set;}

MathMLTreeBar

Shows or hides the Standard toolbar for the "MathML Tree" editing option.

public bool MathMLTreeBar{get; set;}

MathMLTreePropertyBar

Shows or hides the Property pane for the "MathML Tree" editing option.

public bool MathMLTreePropertyBar{get; set;}

MathMLTextBar

Shows or hides the Standard toolbar for the “MathML Text” editing option.

public bool MathMLTextBar{get; set;}

XHTMLBar

Shows or hides the Standard toolbar for the “XHTML” editing option.

public bool XHTMLBar{get; set;}

MathMLMath

Gets or sets the option how to render the <math> element in XHTML (0 — ‘block’, 1 — ‘inline’ mode).

public long MathMLMath {get; set;}

SymbolName4Unicode

Gets or sets the option that defines whether Formulator should try to find the equivalent entity name for Unicode characters while converting expressions into MathML (otherwise the corresponding number will be used).

public bool SymbolName4Unicode{get; set;}

DisplayTimes

Gets or sets the option how to render the <times/> element in the Content markup. It can be rendered using such presentation elements as: 0 — "&times;", 1 — "&sdot;", and 2 — "&InvisibleTimes;".

public long DisplayTimes {get; set;}

ShowInvisibleElements

Gets or sets the option whether to render invisible Content MathML elements.

public bool ShowInvisibleElements{get; set;}

ShowNavigationInfo

Gets or sets the option whether Formulator should display navigation information about the current and the parent node in the Status Bar..

public bool ShowNavigationInfo {get; set;}

ShowRefreshMahML

Gets or sets the option whether Formulator should display notification to a user when the current document needs to be refreshed through MathML. This feature is needed when changing of an option influences all or part of text presentation. E.g., it could be the case of using another symbol for <times/> element of the Content markup, or beautifying of such a formula in Content markup that uses additional slots for inputting <bvar> elements..

public bool ShowRefreshMahML{get; set;}

MultiViewBar

Hides or shows the list of all existing editing modes.

public bool MultiViewBar{get; set;}

ContextInputMode

Switches between Content and Presentation MathML Input Mode (the last one is used by default). 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 Presentation MathML Input Mode leads to inserting of the <mo> element (Presentation markup); in the Content MathML Input Mode such an action inserts a mathematical template (▒+▒) for the <apply> element with the operator element <plus/>.

public bool ContextInputMode{get; set;}

CurrentView

Gets or sets the current editing option (0 – Expression, 1 – MathML Tree, 2 – MathML Text, 3 – XHTML).

public long CurrentView {get; set;}

EnableContextMenu

Control if context menu is enabled.

public bool EnableContextMenu {get; set;}

SimplifiedContextMenu

Switches a context menu to the simplified form (edit commands Cut, Copy, Paste, etc.).

public bool SimplifiedContextMenu {get; set;}

FORMULATOR ACTIVEX CONTROL METHODS

The methods of the Formulator ActiveX Control are listed below.

Methods usage in the free version of the component edition (Formulator Express ActiveX Control) is restricted: only *GetMathML()*, *GetMathMLStr()*, *RelayMessage()*, *GetFormulatorLicenseInfo()* are accessible.

Method Name	Method Description
Copy	Copies the current selection to the clipboard.
Cut	Cuts the current selection to the clipboard.
Paste	Pastes the current clipboard string content at the current caret location.
Undo	Undoes the last action.
Redo	Redoes the last action.
SelectAll	Selects the entire equation.
DialogOtherStyle	Opens the “Other Style” dialog
DialogDefineStyle	Opens the “Define Style” dialog
DialogOtherSize	Opens the “Other Size” dialog
DialogDefineSize	Opens the “Define Size” dialog
SetStyleFaceName	Allows to change the font name assigned to each style.
GetStyleFaceName	Allows to get the font name assigned to each style.
SetStyleBold	Allows to change the weight of the font assigned to each style.
GetStyleBold	Allows to get the weight of the font assigned to each style.
SetStyleItalic	Allows to change the italic property of the font assigned to each style.
GetStyleItalic	Allows to get the italic property of the font assigned to each style.
SetStyleColor	Allows to change the color of the font assigned to each style.
GetStyleColor	Allows to get the color of the font assigned to each style.
SetExpressionColor	Allows to change the color of entire mathematical expression.
SetSymbolSize	Sets the font size in points (pt) for predefined types of text.
GetSymbolSize	Gets the font size in points (pt) for predefined types of text.
SetStyle	Assigns a font and character style and color to selected text or subsequently typed characters (by means of enumerated values).
GetStyle	Retrieves the current style value (as an enumerated

	values).
SetSizeLargeOp	This method assigns a font size of the reserved "Large Symbol" type to selected text or subsequently typed characters.
SetSizeRegular	This method assigns a font size of the reserved "Regular Text" type to selected text or subsequently typed characters.
SetSizeSubscript	This method assigns a font size of the reserved "Subscript" type to selected text or subsequently typed characters.
SetSizeSubSubscript	This method assigns a font size of the reserved "Sub-Subscript" type to selected text or subsequently typed characters.
SetSizeSmaller	This command decreases a font size of the selected text by the number of point that is defined by the argument.
SetSizeLarger	This command increases a font size of the selected text by the number of point that is defined by the argument.
SetMathML	Imports the equation by the MathML 2.0 text storing in the argument.
GetMathML	Exports the current version of the equation that you're working on by saving it in the argument as MathML 2.0 text.
GetMathMLStr	Exports the current version of the equation that you're working on by returning it.
ExportEMF	Exports the current version of the equation that you're working on by saving it on disk into EMF (Enhanced Windows Metafile) *file*.
ExportHEMF	Exports the current version of the equation in the form of *handle in memory* to EMF (Enhanced Windows Metafile).
ExportImage	Exports the current version of the equation that you're working on by saving it on disk in a graphic *file*.
ExportHBitmap	Exports the current version of the equation in the form of *handle in memory* to Bitmap.
ExportXHTML	Exports the current version of the equation for publishing mathematics on web by saving it on disk into XHTML file.
DrawExpression	Draws the current version of the equation that you're working into the Device Context.
RelayMessage	Translates keyboard messages from a container application to the Formulator ActiveX Control.
GetFormulatorLicenseInfo	Returns status of the current version of Formulator

	ActiveX Control.
EmulateToolbarButton	Emulates click on a toolbar's button.
SetRichUIMode	Switches between full and simplified edit modes (the last looks like an usual edit box).
IsRichUIMode	Returns the current edit mode (full or simplified).
Events	
OnChange	Fires when document is changed.
OnEnter	Fires on ENTER
OnChar	Fires on character pressed
OnSetFocus	Fires on set focus
OnKillFocus	Fires on kill focus
OnLMouseClick	Fires on left mouse click
OnLMouseDblClick	Fires on left mouse double click
OnRMouseClick	Fires on right mouse click
OnRMouseDblClick	Fires on right mouse double click

Copy

Copies whatever is currently selected to the Clipboard. Equation is placed on the clipboard in two formats: binary and textual MathML format. Depending on the application that is accessing to this data, one of these copies will be used automatically.

public void Copy();

Cut

Copies whatever is currently selected to the Clipboard, and deletes it from the equation. Equation is placed on the clipboard in two formats: binary and textual MathML format. Depending on the application that is accessing to this data, one of these copies will be used automatically.

public void Cut();

Paste

Inserts the contents of the Clipboard into the equation at the insertion point, or replaces whatever is selected with the contents of the Clipboard. For this command to be successful, the Clipboard must contain a Formulator equation or a text. If the text doesn't meets requirements of the MathML format, it will be inserted literally.

public void Paste();

Undo

Undoes the last command as shown in the text of the menu item. Every command can be undone, back to the point at which the window was opened. This item will be grayed out when there are no more commands to be undone.

public void Undo();

Redo

Redoes the most recent Undo command. Every Undone command can be redone.

public void Redo();

SelectAll

Selects the entire equation, including portions that may be outside the bounds of the window. This command is useful for subsequently copying the equation to the Clipboard for transfer to a word processing document, or before using the Backspace or Delete keys to delete the entire contents of the equation window.

public void SelectAll();

DialogOtherStyle

Opens the “Other Style” dialog, allowing you to assign a font and character style and color to selected text or subsequently typed characters.

public void DialogOtherStyle();

DialogDefineStyle

Opens the “Define Style” dialog, allowing you to change the font and character style and color assigned to each style.

public void DialogDefineStyle();

DialogOtherSize

Opens the “Other Size” dialog, allowing you to assign a font size to selected text or subsequently typed characters.

public void DialogOtherSize();

DialogDefineSize

Opens the "Define Size" dialog, allowing you to change default characters sizes (regular, subscripts, etc.).

public void DialogDefineSize();

SetStyleFaceName
GetStyleFaceName

These methods allows to set/get the font name assigned to each style.

HRESULT SetStyleFaceName([in] LONG nStyle, [in] BSTR bstrMathML);
HRESULT GetStyleFaceName([in] LONG nStyle, [out] BSTR pbstrMathML);*

Method accepts one argument of type *long* that defines a new style and one argument for the font name.

Each character in a Formulator equation can be directly assigned a specific font and character style or can be of one of eleven styles. Each styles is defined as a combination of a font and character style. By changing the style of a text fragment, a user can quickly define its appearance and behavior rules and by changing the definition of a style, a user can quickly change the appearance of all the characters that use it.

The 11 styles available in Formulator are Text, Variable, Function, Greek, Vector-Matrix, Number, Fixed, Operator, Extra-Math, User 1, and User 2. Formulator assigns styles to certain kinds of characters automatically, based on its knowledge of mathematics and typesetting conventions. This intelligent assignment of styles is a useful feature of Formulator which can significantly simplify your work. Assigning style Math to a set of characters a user can explicitly define this intelligent behavior to be a rule for this characters.

```
#define STYLE_MATH          0
#define STYLE_TEXT          1
#define STYLE_VARIABLE      2
#define STYLE_FUNCTION      3
#define STYLE_GREEK         4
#define STYLE_VECTOR        5
#define STYLE_NUMBER        6
#define STYLE_FIXED         7
#define STYLE_OPERATOR      8
#define STYLE_EXTRAMATH     9
#define STYLE_USER1         10
#define STYLE_USER2         11
```

SetStyleBold
GetStyleBold

These methods allows to set/get the weight of the font assigned to each style. Meaning of arguments is the same as for SetStyleFaceName/ GetStyleFaceName.

HRESULT SetStyleBold([in] LONG nStyle, [in] LONG nValue);
HRESULT GetStyleBold([in] LONG nStyle, [out] LONG pnValue);*

SetStyleItalic
GetStyleItalic

These methods allows to set/get the italic property of the font assigned to each style. Meaning of arguments is the same as for SetStyleFaceName/ GetStyleFaceName.

HRESULT SetStyleItalic([in] LONG nStyle, [in] LONG nValue);
HRESULT GetStyleItalic([in] LONG nStyle, [out] LONG pnValue);*

SetStyleColor
GetStyleColor

These methods allows to set/get the color of the font assigned to each style. Meaning of arguments is the same as for SetStyleFaceName/ GetStyleFaceName.

HRESULT SetStyleColor([in] LONG nStyle, [in] OLE_COLOR clrValue);
HRESULT GetStyleColor([in] LONG nStyle, [out] OLE_COLOR pclrValue);*

SetExpressionColor

This method allows to set the color of the entire mathematical expression that is contained in the current document. It applies the given color to all existing styles and graphics (such as, lines, arcs, etc.).

HRESULT SetExpressionColor([in] OLE_COLOR clrValue);

SetSymbolSize
GetSymbolSize

These methods allows to set/get the font size in points (pt) for predefined types of text.

HRESULT SetSymbolSize([in] LONG nType, [in] LONG nValue);
HRESULT GetSymbolSize([in] LONG nType, [out] LONG pnValue);*

Method accepts one argument of type *long* that defines the predefined type of a text and one argument of type *long* for the actual value of font size.

Formulator provides default font sizes for each edit box in a mathematical expression. There are four reserved cases of changing font size:

- large symbols (like sums, products, integrals, etc.) (*nType must be 0*);
- regular text (*nType must be 1*);
- subscript/superscript text (*nType must be 2*);
- subscript/superscript nested in the subscript/superscript text, i.e. subscript/superscript of the next level (*nType must be 3*).

SetStyle

This method assigns a font and character style and color to selected text or subsequently typed characters.

public long SetStyle(long newstyle);

This method accepts one argument of type *long* that defines a new style and returns the old style value.

Each character in a Formulator equation can be directly assigned a specific font and character style or can be of one of eleven styles. Each styles is defined as a combination of a font and character style. By changing the style of a text fragment, a user can quickly define its appearance and behavior rules and by changing the definition of a style, a user can quickly change the appearance of all the characters that use it.

The 11 styles available in Formulator are Text, Variable, Function, Greek, Vector-Matrix, Number, Fixed, Operator, Extra-Math, User 1, and User 2. Formulator assigns styles to certain kinds of characters automatically, based on its knowledge of mathematics and typesetting conventions. This intelligent assignment of styles is a useful feature of Formulator which can significantly simplify your work. Assigning style Math to a set of characters a user can explicitly define this intelligent behavior to be a rule for this characters.

```
#define STYLE_MATH          0
#define STYLE_TEXT          1
#define STYLE_VARIABLE      2
#define STYLE_FUNCTION      3
#define STYLE_GREEK         4
#define STYLE_VECTOR        5
#define STYLE_NUMBER        6
#define STYLE_FIXED         7
#define STYLE_OPERATOR      8
#define STYLE_EXTRAMATH     9
#define STYLE_USER1         10
```

#define STYLE_USER2   11

GetStyle

Retrieves the current style value.

public long GetStyle(void);

The returned style can be one of the above listed numbers (from *STYLE_MATH* to *STYLE_USER2*) or *STYLE_OTHER*:

#define STYLE_OTHER   12

The last style (*STYLE_OTHER*) indicates that a user has selected a new combination of font attributes and text color.

SetSizeLargeOp

This method assigns a font size of the reserved “Large Symbol” type to selected text or subsequently typed characters.

public void SetSizeLargeOp();

SetSizeRegular

This method assigns a font size of the reserved “Regular Text” type to selected text or subsequently typed characters.

public void SetSizeRegular();

SetSizeSubscript

This method assigns a font size of the reserved “Subscript” type to selected text or subsequently typed characters.

public void SetSizeSubscript();

SetSizeSubSubscript

This method assigns a font size of the reserved “Sub-Subscript” type to selected text or subsequently typed characters.

public void SetSizeSubSubscript();

SetSizeSmaller

This command decreases a font size of the selected text by the number of point that is defined by the argument.

public void SetSizeSmaller(long delta);

SetSizeLarger

This command increases a font size of the selected text by the number of point that is defined by the argument.

public void SetSizeLarger(long delta);

SetMathML

Imports the equation by the MathML 2.0 text storing in the argument.

public void SetMathML(LPCTSTR strText);

GetMathML

Exports the current version of the equation that you're working on by saving it in the argument as MathML 2.0 text.

public void GetMathML(VARIANT pText);*

GetMathMLStr

Exports the current version of the equation that you're working on by returning it.

public BSTR GetMathMLStr();

ExportEMF

Exports the current version of the equation that you're working on by saving it on disk into EMF (Enhanced Windows Metafile) file.

HRESULT ExportEmf(BSTR bstrPath);

ExportHEMF

Exports the current version of the equation in the form of handle in memory to EMF (Enhanced Windows Metafile).

HRESULT ExportHEmf();

ExportImage

Exports the current version of the equation that you're working on by saving it on disk in a graphic file.

HRESULT ExportImage(BSTR bstrPath);

ExportHBitmap

Exports the current version of the equation in the form of handle in memory to Bitmap.

HRESULT ExportHBitmap();

ExportXHTML

Exports the current version of the equation for publishing mathematics on web by saving it on disk into XHTML file.

HRESULT ExportXHTML(BSTR bstrPath);

DrawExpression

Draws the current version of the equation that you're working into the Device Context.

void DrawExpression(OLE_HANDLE hDC, long nLeft, long nTop, long nWidth, long nHeight, boolean fScale);

Parameters:

hDC – a device context to draw into;
nLeft, nTop, nWidth, nHeight – area in a device context to draw into;
fScale – if the picture should be inscribed into the given area in a device context.

RelayMessage

Translates keyboard messages from a container application to the Formulator ActiveX Control.

public long RelayMessage(long message, long wparam, long lparam);

GetFormulatorLicenseInfo

Returns status of the current version of Formulator ActiveX Control (0 — trial version; 1 — express version; 2 — full version).

long GetFormulatorLicenseInfo();

EmulateToolbarButton

Emulates click on a toolbar's button.

void EmulateToolbarButton(LPCTSTR strButtonName);

A button is identified by its unique name that must be passed to the method (*strButtonName*). The full list of buttons identifiers is listed below.

	@t@Hidden@ – Hidden templates
@box_dash	Box (Ctrl+Shift+T,X)
text /	Division operator
text _	Under bar
text (	Left parenthesis
text)	Right parenthesis
text [	Left square-brackets
text]	Right square-brackets
text {	Left curly-brackets
text }	Right curly-brackets
text <	Left angle brackets
text >	Right angle brackets
text |	Vertical bar
text ||	Double vertical bar
par text ||	Double vertical bar
@lfloor	Left floor brackets
@rfloor	Right floor brackets
@lceil	Left ceil brackets
@rceil	Right ceil brackets
@apply_function	Apply Function
@invisible_times	Invisible Times
@invisible_coma	Invisible Coma
emsp14	Four-Per-Em Space (1/4 EM)
en space	En Space (1/2 EM)
@mspace	En Space (1/2 EM)
medium space	Medium Space (4/18 EM)
negative medium space	Negative Medium Space (4/18 EM)
nbsp space	Non Breaking Space
very thin space	Very Thin Space (1/18 EM)

zero width space	Zero width space
@empty_frame	Empty frame
@mphantom_frame	MPHANTOM frame
@message_frame	Error/warning message frame
@dintegral	Double integral
@tintegral	Triple integral
@cintegral	Contour integral
@aintegral	Area integral
@vintegral	Volume integral
@awintegral	Integral with counter-clockwise
@cwintegral	Integral with clockwise loop
@assign	Assign
@sqrt_menclose	The menclose element rendered in radical notation

@t@Logics@ – Relational and logical symbols

<=	Less-than or equal to (Ctrl+K,,)
>=	Greater-than or equal to (Ctrl+K,.)
<<	Much less-than
>>	Much greater-than
@precedes	Precedes
@succeeds	Succeeds
@normal_subgroup	Contains as normal subgroup
@cnormal_subgroup	Normal subgroup
!=	Not equal to (Ctrl+K,+)
@identical	Identical to (Ctrl+K,=)
~	Tilde operator (Ctrl+K,Alt+~)
@almost_equal	Almost equal to (Ctrl+K,~)
~-	Asymptotically equal to
~=	Approximately equal to
@proportional	Proportional to (Ctrl+K,P)
@therefore	Therefore (Ctrl+Shift+K,T)
@exist	There exists (Ctrl+Shift+K,E)
@forall	For all (Ctrl+Shift+K,A)
!	Not sign (Ctrl+Shift+K,N)
&	Logical and (Ctrl+Shift+K,7)
|	Logical or (Ctrl+Shift+K,\)
@delta_equal	Delta equal to
@estimates	Estimates
@approaches	Approaches the limit

@t@Space@ – Spaces templates

1-point space	1-point space (Ctrl+Alt+Space)
thin space	Thin space (1/6 EM) (Ctrl+Space)
thick space	Thick space (5/18 EM) (Ctrl+Shift+Space)
em space	EM space (Ctrl+K,4)

@t@Operator@ – Operator symbols

@plusminus	Plus-minus sign (Ctrl+Shift+K,=)
@times	Multiplication sign (Ctrl+K,T)
*	Asterik operator (Ctrl+Shift+K,*)
@divide	Division sign (Ctrl+Shift+K,/)
@circle_plus	Circled plus
@circle_times	Circled times
@plus	Addition sign
@minus	Substraction sign
@center_dot	Dot operator (Ctrl+Shift+K,.)
@bullet	Bullet (Ctrl+Shift+K,8)
@lang	Left-pointing angle bracket (Ctrl+Shift+K,<)
@rang	Right-pointing angle bracket (Ctrl+Shift+K,>)

@t@Arrow@ – Arrow symbols

@arrow_h	Left right arrow (Ctrl+K,Alt+Left)
@arrow_r	Rightwards arrow (Ctrl+K,Right)
@arrow_l	Leftwards arrow (Ctrl+K,Left)
@arrow_v	Up down arrow (Ctrl+K,Alt+Up)
@arrow_u	Upwards arrow (Ctrl+K,Up)
@arrow_d	Downwards arrow (Ctrl+K,Down)
@darrow_h	Left right double arrow (Ctrl+K,Alt+Shift+Left)
@darrow_r	Rightwards double arrow (Ctrl+K,Shift+Right)
@darrow_l	Leftwards double arrow (Ctrl+K,Shift+Left)
@darrow_v	Up down double arrow (Ctrl+K,Alt+Shift+Up)
@darrow_u	Upwards double arrow (Ctrl+K,Shift+Up)
@darrow_d	Downwards double arrow (Ctrl+K,Shift+Down)
@aarrow_	North east arrow
@aarrow_	South west arrow
@aarrow_	South east arrow
@aarrow_	North west arrow
@aarrow_rlarr	Rightwards arrow over leftwards arrow
@aarrow_rlhar	Right harpoon over left harpoon
@arrow_map	Rightwards arrow from bar (Ctrl+K,Tab)
@arrow_corner	Downwards arrow with corner leftwards (Ctrl+K,Enter)

@t@Set@ – Set theory symbols

@in	Element of (Ctrl+K,E)
@not_in	Not an element of (Ctrl+K,Shift+E)
@sadd	Union (Ctrl+K,U)
@smul	Intersection (Ctrl+K,X)
@set_in	Subset (Ctrl+K,C)

@set_in_inv	Superset (Ctrl+K,S)
@set_in_eq	Subset of or equal to
@set_in_eq_inv	Superset of or equal to
@set_not_in	Not a subset of (Ctrl+K,Shift+C)
@set_empty	Empty set (Ctrl+K,O)

@t@Const@ – Special constants

@capitaldifferentiald	Capital differential D
@differentiald	Differential D
@partiald	Partial Differential (Ctrl+K,D)
@exponentiale	Exponential e
@imaginaryi	Imaginary i
pi	Greek small letter pi (Ctrl+G,P)
@planck2pi	Planck constant over two pi (Ctrl+K,H)
@infinity	Infinity (Ctrl+K,I)
@lambda_stroke	Latin small letter lambda with stroke (Ctrl+K,L)
@ell	Script small l (Ctrl+Shift+K,L)

@t@Misc@ – Miscellaneous symbols

@weierstrass	Weierstrass elliptic symbol
imag	Fraktur capital I, imaginary part (Ctrl+K,Shift+I)
real	Fraktur capital R, real part (Ctrl+K,Shift+R)
@aleph	Alef symbol (Ctrl+K,A)
@reals	Blackboard-bold capital R, the set of all real numbers (Ctrl+D,Shift+R)
@integers	Blackboard-bold capital Z, the set of all integer numbers (Ctrl+D,Shift+Z)
@complexes	Blackboard-bold capital C, the set of all complex numbers (Ctrl+D,Shift+C)
@rationals	Blackboard-bold capital Q, the set of all rational numbers (Ctrl+D,Shift+Q)
@naturals	Blackboard-bold capital N, the set of all natural numbers (Ctrl+D,Shift+N)
@dagger	Dagger
Delta	Greek capital letter delta (Ctrl+G,Shift+D)
@gradient	Gradient (nabla)
Omega	Greek capital letter omega (Ctrl+G,Shift+W)
@mho	Inverted ohm sign (Ctrl+Shift+K,Shift+O)
@diam	Diamond
@sum	N-ary summation
@product	N-ary product
@coproduct	N-ary coproduct
@integral	Integral
@deg	Degree sign (Ctrl+Shift+K,D)
@angle	Angle (Ctrl+Shift+K,Shift+A)

@angmsd	Measured angle (Ctrl+Shift+K,Alt+A)
@angsph	Spherical angle (Ctrl+Shift+K,Alt+Shift+A)
@bottom	Perpendicular (Ctrl+Shift+K,P)
@parallel	Parallel (Ctrl+Shift+K,I)

@t@GreekLo@ – Greek characters (lowercase)

alpha	Greek small letter alpha (Ctrl+G,A)
beta	Greek small letter beta (Ctrl+G,B)
gamma	Greek small letter gamma (Ctrl+G,G)
delta	Greek small letter delta (Ctrl+G,D)
epsiv	Greek small letter epsilon (Ctrl+G,E)
zeta	Greek small letter zeta (Ctrl+G,Z)
eta	Greek small letter eta (Ctrl+G,H)
theta	Greek small letter theta (Ctrl+G,Q)
iota	Greek small letter iota (Ctrl+G,I)
kappa	Greek small letter kappa (Ctrl+G,K)
lambda	Greek small letter lambda (Ctrl+G,L)
mu	Greek small letter mu (Ctrl+G,M)
nu	Greek small letter nu (Ctrl+G,N)
xi	Greek small letter xi (Ctrl+G,X)
ogr	Greek small letter omicron (Ctrl+G,O)
pi	Greek small letter pi (Ctrl+G,P)
rho	Greek small letter rho (Ctrl+G,R)
sigmav	Greek sigma symbol (Ctrl+G,Shift+V)
sigma	Greek small letter sigma (Ctrl+G,S)
tau	Greek small letter tau (Ctrl+G,T)
upsilon	Greek small letter upsilon (Ctrl+G,U)
phi	Greek small letter phi (Ctrl+G,F)
phiv	Greek phi symbol (Ctrl+G,J)
chi	Greek small letter chi (Ctrl+G,C)
psi	Greek small letter psi (Ctrl+G,Y)
omega	Greek small letter omega (Ctrl+G,W)

@t@GreekUp@ – Greek characters (uppercase)

Agr	Greek capital letter alpha (Ctrl+G,Shift+A)
Bgr	Greek capital letter beta (Ctrl+G,Shift+B)
Gamma	Greek capital letter gamma (Ctrl+G,Shift+G)
Delta	Greek capital letter delta (Ctrl+G,Shift+D)
Egr	Greek capital letter epsilon (Ctrl+G,Shift+E)
Zgr	Greek capital letter zeta (Ctrl+G,Shift+Z)
EEgr	Greek capital letter eta (Ctrl+G,Shift+H)
Theta	Greek capital letter theta (Ctrl+G,Shift+Q)
Igr	Greek capital letter iota (Ctrl+G,Shift+I)
Kgr	Greek capital letter kappa (Ctrl+G,Shift+K)
Lambda	Greek capital letter lambda (Ctrl+G,Shift+L)

Mgr	Greek capital letter mu (Ctrl+G,Shift+M)
Ngr	Greek capital letter nu (Ctrl+G,Shift+N)
Xi	Greek capital letter xi (Ctrl+G,Shift+X)
Ogr	Greek capital letter omicron (Ctrl+G,Shift+O)
Pi	Greek capital letter pi (Ctrl+G,Shift+P)
Rgr	Greek capital letter rho (Ctrl+G,Shift+R)
Sigma	Greek capital letter sigma (Ctrl+G,Shift+S)
Tgr	Greek capital letter tau (Ctrl+G,Shift+T)
Upsi	Greek upsilon with hook symbol
Phi	Greek capital letter phi (Ctrl+G,Shift+F)
KHgr	Greek capital letter chi (Ctrl+G,Shift+C)
Psi	Greek capital letter psi (Ctrl+G,Shift+Y)
Omega	Greek capital letter omega (Ctrl+G,Shift+W)

@t@Derivative@ – Differentiation

@prime	Prime (Ctrl+Alt+')
@prime2	Double prime (Ctrl+")
@d_dt	Differentiation (t as the independent variable)
@dx_dt	Differentiation of vector X (t as the independent variable)
@dxi_dt	Differentiation of vector (t as the independent variable)

@t@Fence@ – Brackets templates

()	Parenthesis or round-brackets (Ctrl+9)
[]	Brackets or square-brackets (Ctrl+[)
{}	Braces or curly-brackets (Ctrl+{)
<>	Angle brackets (Ctrl+,)
@abs	Single vertical bars (Ctrl+T,|)
@norm	Double vertical bars
@floor	Floor brackets
@ceil	Ceiling brackets
(	Left parenthesis
)	Right parenthesis
[	Left brackets (Ctrl+T,[)
]	Right brackets (Ctrl+T,])
{	Left braces (Ctrl+T,{)
}	Right braces (Ctrl+T,})
<	Left angle brackets (Ctrl+T,<)
>	Right angle brackets (Ctrl+T,>)
| left	Left vertical bar
| right	Right vertical bar
|| left	Left double bar
|| right	Right double bar

@t@Fraction@ – Fraction and radical templates

@fraction	Full-size fraction (Ctrl+F)
@diagonal_fraction	Full-size diagonal fraction (Ctrl+/)
/	Slash fraction (Ctrl+T, Alt+/)
@sqrt	Square root (Ctrl+R)
@root	nth root (Ctrl+T,N)
@longdiv	Long division

@t@Script@ – Subscript and superscript templates

@sup	Superscript (Ctrl+H)
@sub	Subscript (Ctrl+L)
@subsup	Superscript and subscript (Ctrl+J)
@presup	Upper prescript
@presub	Lower prescript
@presubsup	Upper and lower prescript
@suppresup	Upper script and prescript
@subpresub	Lower script and prescript
@suppresup_subpresub	Upper and lower script and prescript
@sup_presub	Upper script and lower prescript
@sub_presup	Upper prescript and lower script
@supsub_presub	Upper and lower script, lower prescript
@sub_presupsub	Upper and lower prescript, lower script
@sup_presupsub	Upper and lower prescript, upper script
@supsub_presup	Upper and lower script, upper prescript
@over	Over-script (Ctrl+T, Shift+L)
@under	Under-script (limit) (Ctrl+T, Alt+L)
@underover	Over-script and under-script (Ctrl+T, L)

@t@Sum@ – Summation templates

@sum_nl	Summation with no limits (Ctrl+T,Shift+S)
@sum_ul	Summation with underscript limit (Ctrl+T,Alt+S)
@sum_uol	Summation with underscript and overscript (Ctrl+T,S)
@sum_subl	Summation with subscript limit
@sum_subsupl	Summation with subscript and superscript limit

@t@Integral@ – Integral templates

@integral_nl	Indefinite integral (no limits) (Ctrl+Shift+I,!)
@integral_uol	Definite integral with underscript and overscript limits (Ctrl+Shift+I,S)
@integral_subsupl	Definite integral with subscript and superscript limit (Ctrl+I)
@integral_ul	Definite integral with underscript limit (Ctrl+Shift+I,Alt+S)

@integral_subl	Definite integral with subscript limit (Ctrl+Shift+I,Alt+1)
@dintegral_nl	Double integral with no limits (Ctrl+Shift+I,@)
@dintegral_ul	Double integral with underscript limit (Ctrl+Shift+I,2)
@dintegral_subl	Double integral with subscript limit (Ctrl+Shift+I,Alt+2)
@tintegral_nl	Triple integral with no limits (Ctrl+Shift+I,#)
@tintegral_ul	Triple integral with underscript limit (Ctrl+Shift+I,3)
@tintegral_subl	Triple integral with subscript limit (Ctrl+Shift+I,Alt+3)
@cintegral_nl	Contour integral with no limits (Ctrl+Shift+I,Shift+C)
@cintegral_ul	Contour integral with underscript limit (Ctrl+Shift+I,C)
@cintegral_subl	Contour integral with subscript limit (Ctrl+Shift+I,Alt+C)
@aintegral_nl	Area integral with no limits (Ctrl+Shift+I,Shift+A)
@aintegral_ul	Area integral with underscript limit (Ctrl+Shift+I,A)
@aintegral_subl	Area integral with subscript limit (Ctrl+Shift+I,Alt+A)
@vintegral_nl	Volume integral with no limits (Ctrl+Shift+I,Shift+V)
@vintegral_ul	Volume integral with underscript limit (Ctrl+Shift+I,V)
@vintegral_subl	Volume integral with subscript limit (Ctrl+Shift+I,Alt+V)
@awintegral_nl	Integral with counter-clockwise loop with no limits (Ctrl+Shift+I,Shift+Left)
@awintegral_ul	Integral with counter-clockwise loop with underscript limit (Ctrl+Shift+I,Left)
@awintegral_subl	Integral with counter-clockwise loop with subscript limit (Ctrl+Shift+I,Alt+Left)
@cwintegral_nl	Integral with clockwise loop with no limits (Ctrl+Shift+I,Shift+Right)
@cwintegral_ul	Integral with clockwise loop with underscript limit (Ctrl+Shift+I,Right)
@cwintegral_subl	Integral with clockwise loop with subscript limit (Ctrl+Shift+I,Alt+Right)

@t@Bars@ – Underbar and overbar templates

@tilde	Tilde (Ctrl+^,~)
@hat	Hat (Ctrl+^,6)
@frown	Arc (Ctrl+^,9)
@jointstatus	Joint status (Ctrl+^,J)
@overbar	Over-bar (Ctrl+^,-)
@doverbar	Double over-bar (Ctrl+^,D)

@underbar	Under-bar (Ctrl+^,_)
@dunderbar	Double Under-bar (Ctrl+^,Shift+D)
@raoverbar	Right arrow over-bar (Ctrl+^,Right)
@laoverbar	Left arrow over-bar (Ctrl+^,Left)
@rhoverbar	Right harpoon over-bar (Ctrl+^,Alt+Right)
@daoverbar	Double-headed arrow over-bar (Ctrl+^,Up)
@raunderbar	Right arrow under-bar (Ctrl+^,Shift+Right)
@launderbar	Left arrow under-bar (Ctrl+^,Shift+Left)
@rhunderbar	Right harpoon under-bar (Ctrl+^,Alt+Shift+Right)
@daunderbar	Double-headed arrow under-bar (Ctrl+^,Shift+Up)
@overbrace	Over brace
@underbrace	Under brace
@hormidstrike	Horizontal mid-line strike through (Ctrl+^,Alt+-)
@vermidstrike	Vertical mid-line strike through
@strike	Strike through (Ctrl+^,X)
@strike_blur	Strike-through (bottom-left to upper-right) (Ctrl+^,/)
@strike_tlbr	Strike-through (top-left to bottom-right) (Ctrl+^,\)

@t@Labels@ – Labelled arrow templates

@rauslot	Right arrow with upper text slot (Ctrl+T,Shift+Right)
@ralslot	Right arrow with lower text slot (Ctrl+T,Alt+Right)
@raulslot	Right arrow with upper and lower text slot (Ctrl+T,Right)
@lauslot	Left arrow with upper text slot (Ctrl+T,Shift+Left)
@lalslot	Left arrow with lower text slot (Ctrl+T,Alt+Left)
@laulslot	Left arrow with upper and lower text slot (Ctrl+T,Left)
@dauslot	Double-headed arrow with upper text slot (Ctrl+T,Shift+Up)
@dalslot	Double-headed arrow with lower text slot (Ctrl+T,Alt+Up)
@daulslot	Double-headed arrow with upper and lower text slot (Ctrl+T,Up)
@ddauslot	Double arrow with upper text slot
@ddalslot	Double arrow with lower text slot
@ddaulslot	Double arrow with upper and lower text slot
@ddlsauslot	Large over small arrow with upper text slot
@ddlsalslot	Large over small arrow with lower text slot
@ddlsaulslot	Large over small arrow with upper and lower text slot
@ddslauslot	Small over large arrow with upper text slot
@ddslalslot	Small over large arrow with lower text slot
@ddslaulslot	Small over large arrow with upper and lower text slot
@hddauslot	Harpoons with upper text slot

@hddalslot	Harpoons with lower text slot
@hddaulslot	Harpoons with upper and lower text slot
@hddlsauslot	Large over small harpoon with upper text slot
@hddlsalslot	Large over small harpoon with lower text slot
@hddlsaulslot	Large over small harpoon with upper and lower text slot
@hddslauslot	Small over large harpoon with upper text slot
@hddslalslot	Small over large harpoon with lower text slot
@hddslaulslot	Small over large harpoon with upper and lower text slot

@t@Products@ – Products and set theory templates

@product_nl	Product with no limits (Ctrl+T,Shift+P)
@product_ul	Product with underscript limit (Ctrl+T,Alt+P)
@product_uol	Product with underscript and overscript limits (Ctrl+T,P)
@product_subl	Product with subscript limit
@product_subsupl	Product with subscript and superscript limit
@coproduct_nl	Coproduct with no limits (Ctrl+T,Shift+C)
@coproduct_ul	Coproduct with underscript limit (Ctrl+T,Alt+C)
@coproduct_uol	Coproduct with underscript and overscript limits (Ctrl+T,C)
@coproduct_subl	Coproduct with subscript limit
@coproduct_subsupl	Coproduct with subscript and superscript limit
@intersection_nl	Intersection with no limits (Ctrl+T,Shift+I)
@intersection_ul	Intersection with underscript limit (Ctrl+T,Alt+I)
@intersection_uol	Intersection with underscript and overscript limits (Ctrl+T,I)
@intersection_subl	Intersection with subscript limit
@intersection_subsupl	Intersection with subscript and superscript limit
@union_nl	Union with no limits (Ctrl+T,Shift+U)
@union_ul	Union with underscript limit (Ctrl+T,Alt+U)
@union_uol	Union with underscript and overscript limits (Ctrl+T,U)
@union_subl	Union with subscript limit
@union_subsupl	Union with subscript and superscript limit

@t@Table@ – Table templates

@table 1x1	1 row 1 column frame
@table 1x2	1 row 2 column table
@table 2x1	2 row 1 column table
@table 2x2	2 row 2 column table
@table 1x3	1 row 3 column table
@table 3x1	3 row 1 column table
@table 3x3	3 row 3 column table

@table 1x4	1 row 4 column table
@table 4x1	4 row 1 column table
@table 4x4	4 row 4 column table
@table 1 x ask	Variable-size row table
@table ask x 1	Variable-size column table
@table ask x ask	Variable-size table

@t@Box@ – Box templates

@box_l	Box (left side)
@box_r	Box (right side)
@box_t	Box (top side)
@box_b	Box (bottom side)
@box_lt	Box (top and left sides)
@box_rt	Box (top and right sides)
@box_lb	Box (bottom and left sides)
@box_rb	Box (bottom and right sides)
@box	Box (Ctrl+Shift+T,X)
@box_a	Actuarial notation
@roundedbox	Rounded box
@circle	Circle

@t@Token@ – Token elements

@cn	The cn element is used to specify actual numerical constants
@cn_input	The cn element is used to specify actual numerical constants
@cn_rational	The cn element for a rational constant
@cn_complex_cartesian	The cn element for a complex-cartesian constant
@cn_complex_polar	The cn element for a complex-polar constant
@ci	The ci element is used to name an identifier in an expression
@ci_input	The ci element is used to name an identifier in an expression
@csymbol	The csymbol element is used to create an element whose semantics are externally defined

@t@Apply@ – Basic Content Elements

@apply2	The apply element allows a function or operator to be applied to its arguments
@apply3	The apply element allows a function or operator to be applied to its arguments
@apply4	The apply element allows a function or operator to be applied to its arguments
@apply_ask	The apply element allows a function or operator to

	be applied to its arguments
@apply_inverse	The functional inverse
@condition	The condition element is used to assert that a Boolean valued expression should be true
@declare_ask	Declare a specific mathematical object
@lambda1	The lambda element is used to construct a user-defined function from an expression, bound variables, and qualifiers
@lambda2	The lambda element is used to construct a user-defined function from an expression, bound variables, and qualifiers
@lambda3	The lambda element is used to construct a user-defined function from an expression, bound variables, and qualifiers
@compose	The compose element represents the function composition operator
@compose2	The compose element represents the function composition operator
@compose3	The compose element represents the function composition operator
@compose_ask	The compose element represents the function composition operator
@compose_q1	The compose element represents the function composition operator
@ident	The ident element represents the identity function
@domain	The domain element denotes the domain of a given function, which is the set of values over which it is defined
@domain_apply	The domain element denotes the domain of a given function, which is the set of values over which it is defined
@codomain	The codomain element denotes the codomain of a given function, which is a set containing all values taken by the function
@codomain_apply	The codomain element denotes the codomain of a given function, which is a set containing all values taken by the function
@image	The image element denotes the image of a given function, which is the set of values taken by the function
@domainofapplication	The domainofapplication element is a qualifier which denotes the domain over which a given function is being applied
@interval_o	The interval element is used to represent simple mathematical intervals of the real number line
@interval_c	The interval element is used to represent simple

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

@t@Piecewise@ – Piecewise Declaration

@piecewise1_	The piecewise element constructs 'piecewise' declarations of functions
@piecewise2_	The piecewise element constructs 'piecewise' declarations of functions
@piecewise3_	The piecewise element constructs 'piecewise' declarations of functions
@piecewise1_ask	The piecewise element constructs 'piecewise' declarations of functions
@piecewise1_1	The piecewise element constructs 'piecewise' declarations of functions
@piecewise2_1	The piecewise element constructs 'piecewise' declarations of functions
@piecewise3_1	The piecewise element constructs 'piecewise' declarations of functions
@piecewise1_ask_1	The piecewise element constructs 'piecewise' declarations of functions
@piecewise	The piecewise element constructs 'piecewise' declarations of functions
@piece	The piece element contains exactly two children: the first child defines the value taken by the piecewise expression when the condition specified in the associated second child of the piece is true
@otherwise	The otherwise element allows the specification of a value to be taken by the piecewise function when none of the conditions (second child elements of the piece elements) is true

@t@Arithmetic@ – Arithmetic Operators

@cm_quotient	The quotient element is the operator used for division modulo a particular base
@cm_quotient2	The quotient element is the operator used for division modulo a particular base
@cm_factorial	The factorial element is used to construct factorials
@cm_factorial1	The factorial element is used to construct factorials
@cm_divide	The division operator
@cm_divide2	The division operator
@cm_minus	The subtraction operator
@cm_minus1	The subtraction operator

@cm_minus2	The subtraction operator
@cm_plus	The addition operator
@cm_plus1	The addition operator
@cm_plus2	The addition operator
@cm_plus3	The addition operator
@cm_plus_ask	The addition operator
@cm_plus_q1	The addition operator
@cm_plus_q2	The addition operator
@cm_plus_q3	The addition operator
@cm_times	The multiplication operator
@cm_times2	The multiplication operator
@cm_times3	The multiplication operator
@cm_times_ask	The multiplication operator
@cm_times_q1	The multiplication operator
@cm_times_q2	The multiplication operator
@cm_times_q3	The multiplication operator
@cm_rem	The remainder of a division modulo a particular base
@cm_rem2	The remainder of a division modulo a particular base
@cm_power	The power element is a generic exponentiation operator
@cm_power2	The power element is a generic exponentiation operator
@cm_root	The root element is used to construct roots
@cm_root2	The square root
@cm_root3	The cube root
@cm_root_any	The root element; the kind of root to be taken is specified by a 'degree' element
@cm_gcd	The greatest common divisor of its arguments
@cm_gcd1	The greatest common divisor of its arguments
@cm_gcd2	The greatest common divisor of its arguments
@cm_gcd3	The greatest common divisor of its arguments
@cm_gcd_ask	The greatest common divisor of its arguments
@cm_lcm	The lowest common multiple of its arguments
@cm_lcm1	The lowest common multiple of its arguments
@cm_lcm2	The lowest common multiple of its arguments
@cm_lcm3	The lowest common multiple of its arguments
@cm_lcm_ask	The lowest common multiple of its arguments
@cm_floor	The round-down (towards -infinity) operator
@cm_floor1	The round-down (towards -infinity) operator
@cm_ceiling	The round-up (towards +infinity) operator
@cm_ceiling1	The round-up (towards +infinity) operator

@t@Algebra@ – Algebra Operators

@cm_abs	The abs element represents the absolute value of a real quantity or the modulus of a complex quantity

@cm_abs1	The abs element represents the absolute value of a real quantity or the modulus of a complex quantity
@cm_conjugate	The conjugate element represents the complex conjugate of a complex quantity
@cm_conjugate1	The conjugate element represents the complex conjugate of a complex quantity
@cm_arg	The arg operator gives the 'argument' of a complex number, which is the angle (in radians) it makes with the positive real axis
@cm_arg1	The arg operator gives the 'argument' of a complex number, which is the angle (in radians) it makes with the positive real axis
@cm_real	The real operator gives the real part of a complex number
@cm_real1	The real operator gives the real part of a complex number
@cm_imaginary	The imaginary operator gives the imaginary part of a complex number
@cm_imaginary1	The imaginary operator gives the imaginary part of a complex number

@t@CMLogic@ – Logic Operators

@cm_and	The and element is the Boolean 'and' operator
@and2	The and element is the Boolean 'and' operator
@and3	The and element is the Boolean 'and' operator
@and_ask	The and element is the Boolean 'and' operator
@cm_and_q1	The and element is the Boolean 'and' operator
@cm_and_q2	The and element is the Boolean 'and' operator
@cm_and_q3	The and element is the Boolean 'and' operator
@cm_or	The or element is the Boolean 'or' operator
@or2	The or element is the Boolean 'or' operator
@or3	The or element is the Boolean 'or' operator
@or_ask	The or element is the Boolean 'or' operator
@cm_or_q1	The or element is the Boolean 'or' operator
@cm_or_q2	The or element is the Boolean 'or' operator
@cm_or_q3	The or element is the Boolean 'or' operator
@cm_xor	The xor element is the Boolean 'exclusive or' operator
@xor2	The xor element is the Boolean 'exclusive or' operator
@xor3	The xor element is the Boolean 'exclusive or' operator
@xor_ask	The xor element is the Boolean 'exclusive or' operator
@cm_xor_q1	The xor element is the Boolean 'exclusive or' operator
@cm_xor_q2	The xor element is the Boolean 'exclusive or' operator
@cm_xor_q3	The xor element is the Boolean 'exclusive or' operator
@cm_not	The not operator is the Boolean 'not' operator
@cm_not1	The not operator is the Boolean 'not' operator

@cm_implies	The implies element is the Boolean relational operator 'implies'
@cm_implies2	The implies element is the Boolean relational operator 'implies'
@cm_forall	The forall element represents the universal quantifier of logic
@cm_forall_bvar	The forall element represents the universal quantifier of logic
@cm_forall_bvar_ask	The forall element represents the universal quantifier of logic
@cm_forall_bvar_cond	The forall element represents the universal quantifier of logic
@cm_forall_bvar_cond_ask	The forall element represents the universal quantifier of logic
@cm_exists	The exists element represents the existential quantifier of logic
@cm_exists_bvar	The exists element represents the existential quantifier of logic
@cm_exists_bvar_ask	The exists element represents the existential quantifier of logic
@cm_exists_bvar_cond	The exists element represents the existential quantifier of logic
@cm_exists_bvar_cond_ask	The exists element represents the existential quantifier of logic

@t@CMMaxMin@ – Maximum and Minimum

@cm_max	The maximum operator
@cm_max1	The maximum operator
@cm_max2	The maximum operator
@cm_max3	The maximum operator
@cm_max_ask	The maximum operator
@cm_max_bvar	The maximum operator; the elements to be compared are described using bound variables with a condition element and an expression to be maximized
@cm_max_bvar_ask	The maximum operator; the elements to be compared are described using bound variables with a condition element and an expression to be maximized
@cm_min	The minimum operator
@cm_min1	The minimum operator
@cm_min2	The minimum operator
@cm_min3	The minimum operator
@cm_min_ask	The minimum operator
@cm_min_bvar	The minimum operator; the elements to be compared

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

@t@CMRelations@ – Relations

@cm_eq	The eq element is the relational operator 'equals'
@cm_eq2	The eq element is the relational operator 'equals'
@cm_eq3	The eq element is the relational operator 'equals'
@cm_eq_ask	The eq element is the relational operator 'equals'
@cm_neq1	The neq element is the 'not equal to' relational operator
@cm_neq2	The neq element is the 'not equal to' relational operator
@cm_equivalent	The equivalent element is the 'equivalence' relational operator
@cm_equivalent2	The equivalent element is the 'equivalence' relational operator
@cm_equivalent3	The equivalent element is the 'equivalence' relational operator
@cm_equivalent_ask	The equivalent element is the 'equivalence' relational operator
@cm_gt	The gt element is the 'greater than' relational operator
@cm_gt2	The gt element is the 'greater than' relational operator
@cm_gt3	The gt element is the 'greater than' relational operator
@cm_gt_ask	The gt element is the 'greater than' relational operator
@cm_lt	The lt element is the 'less than' relational operator
@cm_lt2	The lt element is the 'less than' relational operator
@cm_lt3	The lt element is the 'less than' relational operator
@cm_lt_ask	The lt element is the 'less than' relational operator
@cm_geq	The geq element is the relational operator 'greater than or equal'
@cm_geq2	The geq element is the relational operator 'greater than or equal'
@cm_geq3	The geq element is the relational operator 'greater than or equal'
@cm_geq_ask	The geq element is the relational operator 'greater than or equal'
@cm_leq	The leq element is the relational operator 'less than or equal'

@cm_leq2 The leq element is the relational operator 'less than or equal'

@cm_leq3 The leq element is the relational operator 'less than or equal'

@cm_leq_ask The leq element is the relational operator 'less than or equal'

@cm_factorof1 The factorof element is the relational operator element on two integers a and b specifying whether one is an integer factor of the other

@cm_factorof2 The factorof element is the relational operator element on two integers a and b specifying whether one is an integer factor of the other

@t@CMCalculus@ – Calculus and Vector Calculus

@cm_int The int element is the operator element for an integral

@cm_int1 The int element is the operator element for an integral

@cm_int2 The int element is the operator element for an integral (a definite integral, using the qualifiers lowlimit, uplimit)

@cm_int3 The int element is the operator element for an integral (specifies the domain of integration with an interval element)

@cm_int4 The int element is the operator element for an integral (specifies the domain of integration with an condition element)

@cm_diff The diff element is the differentiation operator element for functions of a single variable

@cm_diff1 The diff element is the differentiation operator element for functions of a single variable (the derivative of a function)

@cm_diff2 The diff element is the differentiation operator element for functions of a single variable (the derivative of an expression with respect to a bound variable)

@cm_diff3 The diff element is the differentiation operator element for functions of a single variable (the derivative of an expression with respect to a bound variable)

@cm_partialdiff The partialdiff element is the partial differentiation operator element for functions or algebraic expressions in several variables

@cm_partialdiff1 The partialdiff element is the partial differentiation operator element for functions or algebraic

	expressions in several variables
@cm_partialdiff2	The partialdiff element is the partial differentiation operator element for functions or algebraic expressions in several variables
@cm_partialdiff3	The partialdiff element is the partial differentiation operator element for functions or algebraic expressions in several variables
@cm_partialdiff1_ask	The partialdiff element is the partial differentiation operator element for functions or algebraic expressions in several variables
@cm_partialdiff2_ask	The partialdiff element is the partial differentiation operator element for functions or algebraic expressions in several variables
@cm_partialdiff3_ask	The partialdiff element is the partial differentiation operator element for functions or algebraic expressions in several variables
@cm_partialdiff4_list1	The partialdiff element is the partial differentiation operator element for functions or algebraic expressions in several variables
@cm_partialdiff4_list2	The partialdiff element is the partial differentiation operator element for functions or algebraic expressions in several variables
@cm_partialdiff4_list_ask	The partialdiff element is the partial differentiation operator element for functions or algebraic expressions in several variables
@cm_divergence	The divergence element is the vector calculus divergence operator
@cm_divergence1	The divergence element is the vector calculus divergence operator
@cm_grad	The grad element is the vector calculus gradient operator
@cm_grad1	The grad element is the vector calculus gradient operator
@cm_curl	The curl element is the vector calculus curl operator
@cm_curl1	The curl element is the vector calculus curl operator
@cm_laplacian	The laplacian element is the vector calculus laplacian operator
@cm_laplacian1	The laplacian element is the vector calculus laplacian operator

@t@CMSet@ – Theory of Sets

@cm_set1	The set element is the container element that constructs a set of elements
@cm_set2	The set element is the container element that constructs a set of elements

@cm_set3	The set element is the container element that constructs a set of elements
@cm_set_ask	The set element is the container element that constructs a set of elements
@cm_set_bvar	The set element is the container element that constructs a set of elements
@cm_set_bvar_ask	The set element is the container element that constructs a set of elements
@cm_list1	The list element is the container element that constructs a list of elements
@cm_list2	The list element is the container element that constructs a list of elements
@cm_list3	The list element is the container element that constructs a list of elements
@cm_list_ask	The list element is the container element that constructs a list of elements
@cm_list_bvar	The list element is the container element that constructs a list of elements
@cm_list_bvar_ask	The list element is the container element that constructs a list of elements
@cm_union	The union element is the operator for a set-theoretic union or join of sets
@cm_union2	The union element is the operator for a set-theoretic union or join of sets
@cm_union3	The union element is the operator for a set-theoretic union or join of sets
@cm_union_ask	The union element is the operator for a set-theoretic union or join of sets
@cm_union_q1	The union element is the operator for a set-theoretic union or join of sets (qualifier elements: lowlimit and uplimit)
@cm_union_q2	The union element is the operator for a set-theoretic union or join of sets (qualifier element: interval)
@cm_union_q3	The union element is the operator for a set-theoretic union or join of sets (qualifier element: condition)
@cm_intersect	The intersect element is the operator for the set-theoretic intersection or meet of sets
@cm_intersect2	The intersect element is the operator for the set-theoretic intersection or meet of sets
@cm_intersect3	The intersect element is the operator for the set-theoretic intersection or meet of sets
@cm_intersect_ask	The intersect element is the operator for the set-theoretic intersection or meet of sets
@cm_intersect_q1	The intersect element is the operator for the set-theoretic intersection or meet of sets (qualifier elements: lowlimit and uplimit)

@cm_intersect_q2	The intersect element is the operator for the set-theoretic intersection or meet of sets (qualifier element: interval)
@cm_intersect_q3	The intersect element is the operator for the set-theoretic intersection or meet of sets (qualifier element: condition)
@cm_in	The in element is the relational operator used for a set-theoretic inclusion
@cm_in2	The in element is the relational operator used for a set-theoretic inclusion
@cm_notin	The notin element is the relational operator element used for set-theoretic exclusion
@cm_notin2	The notin element is the relational operator element used for set-theoretic exclusion
@cm_subset	The subset element is the relational operator element for a set-theoretic containment
@cm_subset2	The subset element is the relational operator element for a set-theoretic containment
@cm_subset_q1	The subset element is the relational operator element for a set-theoretic containment (qualifier elements: lowlimit and uplimit)
@cm_subset_q2	The subset element is the relational operator element for a set-theoretic containment (qualifier element: interval)
@cm_subset_q3	The subset element is the relational operator element for a set-theoretic containment (qualifier element: condition)
@cm_prsubset	The prsubset element is the relational operator element for set-theoretic proper containment
@cm_prsubset2	The prsubset element is the relational operator element for set-theoretic proper containment
@cm_prsubset_q1	The prsubset element is the relational operator element for set-theoretic proper containment
@cm_prsubset_q2	The prsubset element is the relational operator element for set-theoretic proper containment
@cm_prsubset_q3	The prsubset element is the relational operator element for set-theoretic proper containment
@cm_notsubset	The notsubset element is the relational operator element for the set-theoretic relation 'is not a subset of'
@cm_notsubset2	The notsubset element is the relational operator element for the set-theoretic relation 'is not a subset of'
@cm_notprsubset	The notprsubset element is the operator element for the set-theoretic relation 'is not a proper subset of'
@cm_notprsubset2	The notprsubset element is the operator element for

	the set-theoretic relation 'is not a proper subset of'
@cm_setdiff	The setdiff element is the operator element for a set-theoretic difference of two sets
@cm_setdiff2	The setdiff element is the operator element for a set-theoretic difference of two sets
@cm_card	The card element is the operator element for the size or cardinality of a set
@cm_card1	The card element is the operator element for the size or cardinality of a set
@cm_cartesianproduct	The cartesianproduct element is the operator element for the Cartesian product of two or more sets
@cm_cartesianproduct2	The cartesianproduct element is the operator element for the Cartesian product of two or more sets
@cm_cartesianproduct3	The cartesianproduct element is the operator element for the Cartesian product of two or more sets
@cm_cartesianproduct_ask	The cartesianproduct element is the operator element for the Cartesian product of two or more sets
@cm_cartesianproduct_q1	The cartesianproduct element is the operator element for the Cartesian product of two or more sets
@cm_cartesianproduct_q2	The cartesianproduct element is the operator element for the Cartesian product of two or more sets
@cm_cartesianproduct_q3	The cartesianproduct element is the operator element for the Cartesian product of two or more sets

@t@CMSum@ – Sequences and Series

@cm_sum	The sum element denotes the summation operator
@cm_sum1	The sum element denotes the summation operator
@cm_sum_q1	The sum element denotes the summation operator
@cm_sum_q2	The sum element denotes the summation operator
@cm_sum_q3	The sum element denotes the summation operator
@cm_sum_q4	The sum element denotes the summation operator
@cm_product	The product element denotes the product operator
@cm_product1	The product element denotes the product operator
@cm_product_q1	The product element denotes the product operator
@cm_product_q2	The product element denotes the product operator
@cm_product_q3	The product element denotes the product operator
@cm_product_q4	The product element denotes the product operator
@cm_limit	The limit element represents the operation of taking a limit of a sequence
@cm_limit1	The limit element represents the operation of taking a limit of a sequence
@cm_limit_q1	The limit element represents the operation of taking a limit of a sequence
@cm_limit_q2	The limit element represents the operation of taking a limit of a sequence

@cm_limit_q3	The limit element represents the operation of taking a limit of a sequence
@cm_limit_q4	The limit element represents the operation of taking a limit of a sequence
@cm_tendsto	The tendsto element is used to express the relation that a quantity is tending to a specified value
@cm_tendsto	The tendsto element is used to express the relation that a quantity is tending to a specified value
@cm_tendsto	The tendsto element is used to express the relation that a quantity is tending to a specified value
@cm_tendsto	The tendsto element is used to express the relation that a quantity is tending to a specified value
@cm_tendsto	The tendsto element is used to express the relation that a quantity is tending to a specified value
@cm_tendsto	The tendsto element is used to express the relation that a quantity is tending to a specified value

@t@Trigonometric@ – Common Trigonometric Functions

@sin	The standard trigonometric function: sin
@cos	The standard trigonometric function: cos
@tan	The standard trigonometric function: tan
@sec	The standard trigonometric function: sec
@csc	The standard trigonometric function: csc
@cot	The standard trigonometric function: cot
@arcsin	The standard trigonometric function: arcsin
@arccos	The standard trigonometric function: arccos
@arctan	The standard trigonometric function: arctan
@arcsec	The standard trigonometric function: arcsec
@arccsc	The standard trigonometric function: arccsc
@arccot	The standard trigonometric function: arccot
@sin_apply	The standard trigonometric function: sin
@cos_apply	The standard trigonometric function: cos
@tan_apply	The standard trigonometric function: tan
@sec_apply	The standard trigonometric function: sec
@csc_apply	The standard trigonometric function: csc
@cot_apply	The standard trigonometric function: cot
@arcsin_apply	The standard trigonometric function: arcsin
@arccos_apply	The standard trigonometric function: arccos
@arctan_apply	The standard trigonometric function: arctan
@arcsec_apply	The standard trigonometric function: arcsec
@arccsc_apply	The standard trigonometric function: arccsc
@arccot_apply	The standard trigonometric function: arccot

@t@Hyperbolic@ – Common Hyperbolic Functions

@sinh	The standard hyperbolic function: sinh

@cosh	The standard hyperbolic function: cosh
@tanh	The standard hyperbolic function: tanh
@sech	The standard hyperbolic function: sech
@csch	The standard hyperbolic function: csch
@coth	The standard hyperbolic function: coth
@arcsinh	The standard hyperbolic function: arcsinh
@arccosh	The standard hyperbolic function: arccosh
@arctanh	The standard hyperbolic function: arctanh
@arcsech	The standard hyperbolic function: arcsech
@arccsch	The standard hyperbolic function: arccsch
@arccoth	The standard hyperbolic function: arccoth
@sinh_apply	The standard hyperbolic function: sinh
@cosh_apply	The standard hyperbolic function: cosh
@tanh_apply	The standard hyperbolic function: tanh
@sech_apply	The standard hyperbolic function: sech
@csch_apply	The standard hyperbolic function: csch
@coth_apply	The standard hyperbolic function: coth
@arcsinh_apply	The standard hyperbolic function: arcsinh
@arccosh_apply	The standard hyperbolic function: arccosh
@arctanh_apply	The standard hyperbolic function: arctanh
@arcsech_apply	The standard hyperbolic function: arcsech
@arccsch_apply	The standard hyperbolic function: arccsch
@arccoth_apply	The standard hyperbolic function: arccoth

@t@Exponential@ – Common Exponential Functions

@exp	The exponential function
@ln	The natural logarithm function
@log	The operator that returns a logarithm to a base of 10
@exp_apply	the exponential function
@ln_apply	The natural logarithm function
@log_apply	The operator that returns a logarithm to a given base

@t@CMStatistics@ – Statistics

@cm_mean	mean is the operator element representing a mean or average
@cm_mean1	mean is the operator element representing a mean or average
@cm_mean2	mean is the operator element representing a mean or average
@cm_mean3	mean is the operator element representing a mean or average
@cm_mean_ask	mean is the operator element representing a mean or average
@cm_sdev	sdev is the operator element representing the statistical standard deviation operator

@cm_sdev1	sdev is the operator element representing the statistical standard deviation operator
@cm_sdev2	sdev is the operator element representing the statistical standard deviation operator
@cm_sdev3	sdev is the operator element representing the statistical standard deviation operator
@cm_sdev_ask	sdev is the operator element representing the statistical standard deviation operator
@cm_variance	variance is the operator element representing the statistical variance operator
@cm_variance1	variance is the operator element representing the statistical variance operator
@cm_variance2	variance is the operator element representing the statistical variance operator
@cm_variance3	variance is the operator element representing the statistical variance operator
@cm_variance_ask	variance is the operator element representing the statistical variance operator
@cm_median	median is the operator element representing the statistical median operator
@cm_median1	median is the operator element representing the statistical median operator
@cm_median2	median is the operator element representing the statistical median operator
@cm_median3	median is the operator element representing the statistical median operator
@cm_median_ask	median is the operator element representing the statistical median operator
@cm_mode	mode is the operator element representing the statistical mode operator
@cm_mode1	mode is the operator element representing the statistical mode operator
@cm_mode2	mode is the operator element representing the statistical mode operator
@cm_mode3	mode is the operator element representing the statistical mode operator
@cm_mode_ask	mode is the operator element representing the statistical mode operator
@cm_moment	The moment element represents the statistical moment operator
@cm_moment1	The moment element represents the statistical moment operator
@cm_moment2	The moment element represents the statistical moment operator

@t@CMLinearAlgebra@ – Linear Algebra

@cm_vector 1x1	1 row 1 column vector
@cm_vector 2x1	2 row 1 column vector
@cm_vector 3x1	3 row 1 column vector
@cm_vector 4x1	4 row 1 column vector
@cm_vector ask x 1	Variable-size vector
@cm_matrix 1x1	1 row 1 column matrix
@cm_matrix 1x2	1 row 2 column matrix
@cm_matrix 2x1	2 row 1 column matrix
@cm_matrix 2x2	2 row 2 column matrix
@cm_matrix 1x3	1 row 3 column matrix
@cm_matrix 3x1	3 row 1 column matrix
@cm_matrix 3x3	3 row 3 column matrix
@cm_matrix 1x4	1 row 4 column matrix
@cm_matrix 4x1	4 row 1 column matrix
@cm_matrix 4x4	4 row 4 column matrix
@cm_matrix 1 x ask	Variable-size row matrix
@cm_matrix ask x 1	Variable-size column matrix
@cm_matrix ask x ask	Variable-size matrix
@determinant	The determinant of a matrix
@determinant_apply	The determinant of a matrix
@transpose_apply	The transpose of a matrix
@selector_apply2	Indexing into vectors, matrices and lists
@selector_apply3	Indexing into vectors, matrices and lists
@selector_apply_ask	Indexing into vectors, matrices and lists
@vectorproduct	The operator element for deriving the vector product of two vectors
@vectorproduct_apply	The vector product of two vectors
@scalarproduct	The operator element for deriving the scalar product of two vectors
@scalarproduct_apply	The scalar product of two vectors
@outerproduct	The operator element for deriving the outer product of two vectors
@outerproduct_apply	The outer product of two vectors

@t@CMSemantics@ – Semantics

@cm_semantics	The semantics element associates additional representations with a given MathML construct

@t@CMConstant@ – Constant and Symbol Elements

@integers_cm	The set of all integer numbers
@reals_cm	The set of all real numbers
@rationals_cm	The set of all rational numbers
@naturalnumbers_cm	The set of all natural numbers

@complexes_cm	The set of all complex numbers
@primes_cm	The set of all natural prime numbers
@exponentiale_cm	The exponential base of the natural logarithms
@imaginaryi_cm	The mathematical constant which is the square root of -1
@notanumber_cm	The result of an ill-defined floating point operation
@true	The logical constant for truth
@false	The logical constant for false
@emptyset_cm	The empty set
@pi_cm	The ratio of a circle's circumference to its diameter
@eulergamma_cm	The Euler's constant (approximately 0.5772156649)
@infinity_cm	The concept of infinity

@t@CMQualifier@ – Qualifier Elements

@cm_qualifier_q1_ask	Qualifier elements: lowlimit, uplimit
@cm_qualifier_q2_ask	Qualifier elements: interval
@cm_qualifier_q3_ask	Qualifier elements: condition
@cm_qualifier_q4_ask	Qualifier elements: domainofapplication

SetRichUIMode

Switches between full and simplified edit modes (the last looks like an usual edit box).

void SetRichUIMode(boolean fRich);

IsRichUIMode

Returns the current edit mode (full or simplified).

boolean IsRichUIMode(void);

OnChange

Fires when document is changed.

void OnChange();

OnEnter

Fires on ENTER.

void OnEnter();

OnChar

Fires on character pressed.

void OnChar(ULONG nChar);

OnSetFocus

Fires on set focus.

void OnSetFocus();

OnKillFocus

Fires on kill focus.

void OnKillFocus();

OnLMouseClick

Fires on left mouse click.

void OnLMouseClick(ULONG nFlags, LONG x, LONG y);

OnLMouseDblClick

Fires on left mouse double click.

void OnLMouseDblClick(ULONG nFlags, LONG x, LONG y);

OnRMouseClick

Fires on right mouse click.

void OnRMouseClick(ULONG nFlags, LONG x, LONG y);

OnRMouseDblClick

Fires on right mouse double click.

void OnRMouseDblClick(ULONG nFlags, LONG x, LONG y);