

# ReforMath Library Version 1.1 API Reference

## Disclaimer

Use of the ReforMath library comes with no guarantee or warranty whatsoever. You will be solely responsible for any damage to your system(s) or for any loss of data that may arise from the use of the library. Use at your own risk.

## Overview

ReforMath is library for converting MathML to a variety of formats such as PDF, PNG, SVG, and XAML. It has a C interface but written in C++. The C interface allows it to be used easily with different programming languages. ~~The library is currently available only for Windows; however, a version for Linux or Solaris will be available soon since the source code is almost platform-independent.~~ The library is available for both Windows and Linux.

## License

The ReforMath library is being made available for evaluation purposes only (30-day trial). Your feedback is highly appreciated; spread the word – blog about it, tell your friends/peers about it. If you donate US\$10 – or more – to my PayPal account ([pfranejr@hotmail.com](mailto:pfranejr@hotmail.com)), you may keep the library for personal use (redistribution not allowed; strictly for home/individual use only). For other inquiries (e.g., commercial/site licensing), please send an email to [reformath@hotmail.com](mailto:reformath@hotmail.com) or visit my websites: <http://reformath.webnode.com> and <http://reformath.weebly.com>.

*If the library is well received, I may even make it freeware for personal use and commercial for others. Or if I receive funding from foundations or philanthropists, I may make it open source. Or, if you want to market this software, please contact me. I am also willing to sell the source codes to the willing buyer. Please contact me if interested.*

## Dependencies

STIX fonts version 1.0 (not included). You can get them here: <http://www.stixfonts.org>. ReforMath also requires the Cairo Graphics Library and FreeType, which are all included in the SDK for Windows. They are normally available already under Linux, especially FreeType.

## Limitations

- Fences do not stretch when used with `<mo>`; you must use `<mfenced>`.
  - The underscript and overscript of `<munderover>` are not aligned properly when used with integrals
  - Not all named entities are available.
  - The STIX fonts STIXVariants and STIXNonUnicode are not used.
- All these will be addressed soon.

## Supported Elements and Attributes

ReforMath supports only a subset of MathML 2.0. The table below shows the elements supported and the corresponding attributes (note that even for some elements, not all attributes are supported). If no attribute is given, the attribute is not supported.

Elements/Attributes	Comments
<code>math</code>	
<code>display</code> (block or inline)	This attribute replaces the <code>mode</code> attribute
<code>mode</code> (display or inline)	This attribute is deprecated but still supported by ReforMath
<code>mi</code> , <code>mn</code> , <code>mtext</code>	
<code>mathvariant</code>	All values supported
<code>ms</code>	
<code>mathvariant</code>	All values supported
<code>open/close</code>	N/A. To be added
<code>mo</code>	
<code>lspace/rspace</code>	POSITIVE values only or named spaces such as "veryverythinmathspace", "verythinmathspace", "thinmathspace", "mediummathspace", "thickmathspace", "verythickmathspace", or "veryverythickmathspace"
<code>largeoperator</code>	Ignored. Actual value is derived from the tag content (i.e., if the operator is really a large operator, then the value defaults to 'true'; otherwise, 'false'. So, it is NOT possible to treat a large operator as a non-large operator.
<code>movablelimits</code> (boolean)	This attribute is used only if the tag content is really a large operator with movable limits, such as the summation operator ( $\sum$ ), etc.
<code>accent</code> (boolean)	
<code>stretchy</code> (boolean)	Most stretchable horizontal accents are supported (see <code>munder/mover</code> below). However, fences DO NOT stretch – you must use <code>&lt;mfenced&gt;</code> for stretchy fences.
	Stretchy fences will be supported soon. In the meantime, you have to stick with <code>&lt;mfenced&gt;</code>
<code>mfenced</code>	
<code>open/close</code>	The most common fences are supported: angular brackets

	braces (square) brackets (curly) ceiling floor moustaches parentheses solidus/backslash vertical bars (single/double) These fences can stretch variably. The following, however, cannot stretch beyond the maximum available glyphs (STIXSizeFourSym.otf font) due limited availability of glyph pieces: white brackets (square) white parentheses white angular brackets white braces (curly) tortoise shell
separators	Only the usual separators are recognized such as comma, semicolon, exclamation point.
matrix	
columnalign	Only left, right, center values are supported
columnspacing	Unsigned numeric values or named spaces, see <mo> above
displaystyle (boolean)	
mtr	
mtl	
columnalign	Only left, right, center values are supported
mlabeledtr	NOT SUPPORTED. Unpredictable results.
mfrac	
linethickness	Valid values are 0 to 3 pixels only or its equivalent (e.g., 1.5 pt). Values outside this range will be clamped
bevelled (true or false)	
mroot	
msub	
msubsup	
msup	
munder	
accent-under	Not all accents are supported. &UnderBar; (stretchable) is supported as well as some arrows (also stretchable): left arrow, right arrow, left-right

	arrow, left double arrow, right double arrow, left-right double arrow – These are the only arrows supported by the equation editor of MS Word 2007/2010, to the best of my knowledge.
mover	<p>Not all accents are supported: <math>\overline{\phantom{x}}</math> is supported as well as some arrows (see above). Other accents include hat and tilde. All these are stretchable. The most common short accents are supported such as acute, grave, macron, umlaut, etc.</p> <p>The following are not available but will be supported in a future version: parentheses, brackets, braces</p>
munderover	
mSPACE	POSITIVE values only or named spaces, see above.
mmultiscripts	
mphantom	N/A. To do.
mencloned	N/A. To do.

## Additional limitations

Not all named entities are supported. Preferably, you should use numeric entities. However, support for numeric entities is incomplete, in particular, the fonts STIXVariants and STIXNonUnicode are not used. They will be supported in a future version. At any rate, thousands of glyphs are supported.

## API Descriptions

ReforMath can convert MathML to four different types of files. These types can be selected using the following enumeration:

```
enum mml_file_type
{
    mml_file_type_pdf,
    mml_file_type_png,
    mml_file_type_svg,
    mml_file_type_xaml
};
```

ReforMath uses the following structure to provide information about the image size (in pixels):

```
struct mml_image_metrics
{
    int baseline, width, height, text_height, text_depth;
    double fwidth, fheight;
};
```

where `baseline` is the formula's baseline and is the center of the image (i.e., the image is centered at the baseline), `width` is the image width (this is the rounded value of `fwidth`), `height` is the total height of the image (this is the rounded value of `fheight`), `text_height` is the total ascent of the image or the height of the tallest glyph in the image, `text_depth` is the total descent of the image or the lowest part of the glyph in the image (this value is positive rather than negative), `fwidth` is the width of the image as a double, and `fheight` is the height of the image as a double.

Use `width` and `height` for PNG and `fwidth` and `fheight` for the vector formats.

ReforMath also uses a handle type (`mml_handle`) whose content or content(s) are not given.

The description of each function exported by the library is given below. As a general rule, the functions must be called in the following order (assuming, of course, that no. 1 succeeds):

1. `mml_create_handle()`
2. `mml_convert()`
3. `mml_can_render()`
4. any other functions
5. `mml_free()`

Parameters are mostly of *in* type, unless otherwise specified as *out* or as *in-out*. Also, all functions with a return type of *bool* return either *true* or *false* obviously.

```
LIBREFORMATH_API mml_handle mml_create_handle(void);
```

Creates the handle used by the rest of the functions. This must be called FIRST.

```
LIBREFORMATH_API void mml_free_handle( mml_handle handle );
```

Frees the handle allocated by `mml_create_handle()`. Call this when your application no longer needs the handle since the handle uses a LOT of resources, especially fonts.

```
LIBREFORMATH_API bool mml_convert( mml_handle handle, const char  
*mathml_text );
```

Converts `mathml_text` to something that can be rendered later. The argument `mathml_text` may be either ANSI or UTF-8. It may contain named entities such as `&sum;`, `&alpha;`, etc., but

numeric entities are preferred since the named entities supported by the library are still incomplete.

Also, the root element `<math>` is optional, in which case the output defaults to `'inline'`. It is needed if you wish to create a display formula, e.g., `<math display='block' ...>`.

This function may be called for each MathML formula that you want to convert. The parser will be automatically reset. Alternatively, you may explicitly call `mml_clear()`, but there is no benefit in doing so.

*NOTE: This function will be improved in a future version which will provide more information about errors. So, preferably, the MathML input must be syntactically correct, such as those provided by reliable third-party software, rather than manually created.*

```
LIBREFORMATH_API bool mml_save_to_file( mml_handle handle, const char
*filename, mml_file_type file_type );
```

Saves the output of `mml_convert()` to a file. Even if `mml_convert()` fails, if a portion of the input MathML has been converted successfully, then, at least, *something* can be saved, but this is *not* recommended since the output may be distorted. See `mml_can_render()` below.

To scale the output, call `mml_set_scale()` BEFORE calling this function. See the description of `mml_set_scale()`.

The *filename* parameter may be ANSI or UTF-8. If no extension is explicitly given, NONE will be appended, so you must provide it yourself. Also, the file format is based on the value of *file\_type* rather than on the extension, so make sure that they match.

```
LIBREFORMATH_API bool mml_save_to_file_w( mml_handle handle, const
wchar_t *filename, mml_file_type file_type );
```

The widechar version of `mml_save_to_file()`. See the description above.

```
LIBREFORMATH_API bool mml_save_to_stream( mml_handle handle, [out]
unsigned char **ptr, [out] int *block_size, mml_file_type
file_type );
```

Saves the output of `mml_convert()` to memory rather than to a file. The memory used by *ptr* will be allocated by the library and must be freed with `mml_destroy_stream()`. The parameter *block\_size* is the size in bytes of the allocated stream; this does not include the null terminator if the format is text, such as *SVG* or *XAML*. If the stream contains text, such as *SVG* and *XAML*, the string or stream is null terminated.

To obtain the dimensions of the image, call `mml_get_image_size()`. See below.

```
LIBREFORMATH_API double mml_set_scale( mml_handle handle, double
xy );
```

Sets the x- and y scale of the output. The parameter *xy* cannot be less than 0.5 nor greater than 10.0. If

the value is out of range, it is clamped to this range. The default scale is 1.0.

This function *affects* the result of `mml_save_to_file()`/`mml_save_to_file_w()` and `mml_save_to_stream()`, so it must be called first before calling any of these 3 functions if you want scaling. This function has no effect, however, on `mml_convert()`, since the latter does not perform any rendering.

The function returns the actual value used which may be different from the one supplied if the latter is out of range.

```
LIBREFORMATH_API double mml_get_scale( mml_handle handle );
```

Returns the current scale. The default is 1.0 if the scale has not been set.

```
LIBREFORMATH_API bool mml_can_render( mml_handle handle );
```

Determines whether there is something to render as a result of calling `mml_convert()`. Partial rendering is possible even if there is an error, e.g., if the error in the MathML input is found at the end. However, even with partial rendering, the output may not be desirable, especially in cases of missing elements.

*Note: a number of MathML elements have a fixed number of child elements. If a child element is missing, the internal formatting will also be incomplete and, hence, the output is likely to be distorted.*

To be sure, avoid calling this function if `mml_convert()` fails.

```
LIBREFORMATH_API void mml_clear( mml_handle handle );
```

Resets the parser. After calling this function, `mml_can_render()` will return FALSE.

*Note. The parser is also reset each time `mml_convert()` is called.*

```
LIBREFORMATH_API void mml_destroy_stream( [in-out] unsigned char  
**ptr );
```

Frees the memory allocated by `mml_save_to_stream()`. Note that the parameter is a pointer to a pointer so the function can set *\*ptr* to NULL.

```
LIBREFORMATH_API bool mml_get_image_size( mml_handle handle, [out]  
mml_image_metrics * size );
```

Gets the image size and returns it in the *size* parameter. See the description of `mml_image_metrics` above.

```
LIBREFORMATH_API bool mml_copy_to_clipboard( mml_handle handle, HWND  
hwnd );
```

Copies the converted MathML to the clipboard as a bitmap (i.e., *HBITMAP*).

This function is available only under Windows.

```
LIBREFORMATH_API HBITMAP mml_create_bitmap( mml_handle handle, HDC  
hdc, [out] mml_image_metrics * size );
```

Creates an *HBITMAP* from the converted MathML. *hdc* is the reference HDC (required). The dimensions of the image are returned in *size*; this parameter cannot be NULL.

The function return an *HBITMAP* on success or NULL on failure.

This function is available only under Windows.

---

Copyright © 2011 Peter D. Frane, Jr. All Rights Reserved.

Emails: [reformath@hotmail.com](mailto:reformath@hotmail.com) and [pfranejr@hotmail.com](mailto:pfranejr@hotmail.com) (PayPal)

Websites (temporary): <http://reformath.webnode.com> and <http://reformath.weebly.com>

No part of this document may be reproduced in any form or by any means without the explicit permission of the author. In addition, this document may not be distributed separately from the other files of the SDK. All other trademarks are the property of their respective owners.