

The following steps and the code below show you how easy it is to use the ReforMath library; the comments alone should give you an idea what's going on:

1. Include the library's header 'libreformath.h'
2. Declare an mml_handle variable
3. Call mml_create_handle() to create the handle
4. Call mml_convert() to convert your MathML input
5. Call mml_can_render() to determine whether there is something to render. Even in case of errors, a portion of the input may have been converted successfully
6. Do something with the result by calling mml_save_to_file() (UTF-8) or mml_save_to_file_w() (the wchar version) or mml_save_to_stream(). When saving to memory, you must call mml_destroy_stream() to free any memory allocated
7. Free the handle by calling mml_free()

REMINDER: Please make sure the STIX fonts are installed!

```
#include "libreformath.h" // the library's header (required)
#include <stdio.h>
#include <string>

using namespace std;

#pragma comment( lib, "libreformath.lib" )

int main( int argc, char **argv )
{
    // declare the handle
    mml_handle *mhandle;
    bool result;
    string input;

    // create a handle

    mhandle = mml_create_handle();

    if( !mhandle )
    {
        perror( "unable to create handle" );

        return 1;
    }

    // let's create some MathML input
    // <math> tag is optional; the default is inline; the xmlns attribute is
    ignored

    input = "<math display='block'>";
    input += "<mfenced open='[' close=']'><mfrac>";
    input += "<mn>1</mn><mn>2</mn>";
    input += "</mfrac></mfenced>";
    input += "</math>";

    // convert and check the result

    result = mml_convert( mhandle, input.c_str() );

    if( result && mml_can_render( mhandle ) )
```

```

{
    // call either mml_save_to_file() or mml_save_to_file_w to save to a file
    // or mml_save_to_stream to save to memory. When saving to memory, you must
call
    // mml_destroy_stream() to free any memory used

    // Return values omitted for brevity.
    // It's your responsibility to provide an explicit extension for the
filename.
    // The file type is based on the second argument and NOT on the file
extension!

    mml_save_to_file( mhandle, "sample.pdf", mml_file_type_pdf );
    mml_save_to_file( mhandle, "sample.svg", mml_file_type_svg );
    mml_save_to_file( mhandle, "sample.xaml", mml_file_type_xaml );

    // do some scaling

    mml_set_scale( mhandle, 2.0 );

    mml_save_to_file( mhandle, "sample.png", mml_file_type_png );

    // to save to a stream, declare these 2 variables

    unsigned char *buffer = NULL;
    int buf_size;

    mml_save_to_stream( mhandle, &buffer, &buf_size, mml_file_type_svg );

    // do something with the output like dumping to console

    puts( (char *) buffer );

    // free the buffer; note that the argument is a
    // pointer to a pointer so the buffer can be set to NULL

    mml_destroy_stream( &buffer );
}

// free the handle when done

mml_free( mhandle );

return 0;
}

```

Copyright © 2011 Peter D. Frane, Jr. All Rights Reserved.
 Emails: reformath@hotmail.com and pfranejr@hotmail.com (PayPal)
 Website: <http://reformath.weebly.com> (temporary)

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.