Using MathType to Create TeX and MathML Equations

Paul Topping
Design Science, Inc.
4028 Broadway
Long Beach, CA 90803
USA
pault@mathtype.com
www.mathtype.com

Abstract

MathType 4.0 is the latest release of Design Science's interactive mathematical equation editing software package, the full-featured version of the Equation Editor applet that comes with Microsoft Word. Its completely re-architected translation system should be of particular interest to the TeX and MathML communities. MathType can be used as an aid to learning TeX, as a simpler interface for entering equations into a TeX authoring system, or as part of a document conversion scheme for journal and book publishers.

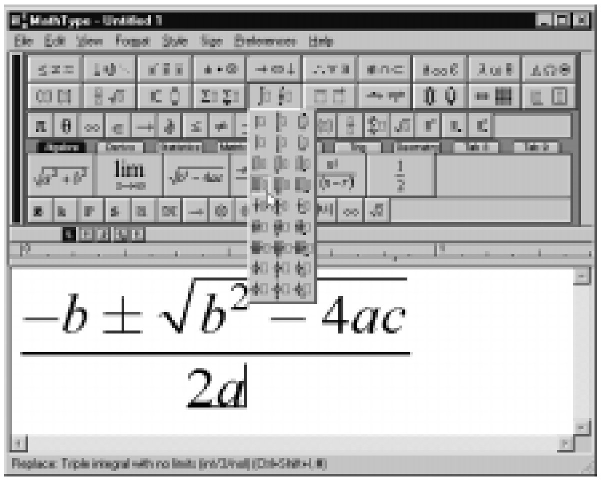
After the introduction, a simple translation example is given to show how its Translator Definition Language (TDL) is used to convert a MathType equation to TeX. This is followed by an introduction to MathML and MathType's MathML translators are discussed. Finally, some of the possibilities for creating translators for special purposes is mentioned, along with discussion on how MathType's translation facilities can be used as component of a more comprehensive document conversion process.

Introduction

MathType is an interactive tool for authoring mathematical material. It runs on Microsoft Windows and Apple Macintosh systems (a Linux implementation is under consideration). Readers may also be familiar with MathType's junior version, Equation Editor, as it is supplied as part of many personal computer software products, such as Microsoft Word and Corel WordPerfect.

Unlike TeX, MathType does not process entire documents. Rather, it is used in conjunction with other products, such as word processors, page layout programs, presentation programs, web/HTML editors, spreadsheets, graphing software, and virtually any other kind of application that allows insertion of a graphical object into its documents. MathType equations can even be inserted into database fields with most modern database systems!

MathType has a simple but powerful direct-manipulation interface for creating standard mathematical notation. Instead of entering a computer language, such as TeX, the MathType user combines simple typing with the insertion of "templates". For example, inserting a fraction template results in a fraction bar with empty slots above and below for the numerator and denominator. The contents of each slot are filled in by the user by more typing and inserting of templates. The displayed equation is reformatted as the user types and spacing is added automatically (although spacing may be explicitly overridden). Some find this interface to be simpler than direct TeX input as there are no keywords to



$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Figure 1: The MathType Window

remember and, most importantly, no possibility of syntax errors.

One common complaint heard from TEX users upon first seeing MathType's user interface is that one must use the mouse for everything. Whereas mouse support is an important part of MathType's user interface (as with virtually all Windows and Mac applications), MathType 4.0 has keyboard methods for performing all of its commands. Also, users may assign keystrokes to commands in any kind of mnemonic scheme they care to invent. The ability to assign a keystroke to an arbitrary math expression is analogous to TEX's macro facility.

Although a thorough examination of MathType is beyond the scope of this paper, here are some of its most important features:

- Any national language characters that the host operating system allows may be inserted into math, including Asian characters.
- MathType's internal representation of characters is Unicode,[1] extended via its Private Use area to cover more of the characters that appear in mathematical notation. We call this MTCode. A user-extendable database of math font-to-MTCode mappings is used to relate characters entered to knowledge used in line formatting, as well as translation.
- A basic set of mathematical fonts (Roman, Greek, italics, Fraktur, blackboard bold, and many mathematical symbols) is included. MathType can also make use of any PostScript Type 1 or TrueType font available via the operating system.
- MathType also includes a translation system for converting mathematics entered in its editing window to virtually any text-based language. This translation system is the chief subject of this paper. In particular, it includes translators for several flavors of TEX and MathML.

MathType's translation facilities

MathType has had a TEX translator for many years that allows the user to copy all or part of an expression onto the clipboard in the TEX language, ready to be pasted into a document. However, until version 4.0, it had two important limitations: it could only generate plain TEX and the user had no control over the TEX fragments generated for particular symbols and templates. MathType 4.0 features a complete re-design of the translator mechanism. The translation of a MathType expression is controlled by a translator definition file, a text file containing a simple translation rule language that allows a fragment of the target language to be associated with each of MathType's many symbols and templates. Although the chief motivation for its development was TEX translation, it can also be used to convert MathType equations to other languages, such as MathML and those specified by the math parts of various SGML-based document languages.

MathType is supplied with translator definition files for plain TEX, AMS-TEX, LATEX, AMS-LATEX, and four MathML variations. These can be customized for specific applications or translators for other mathematical languages can be written by starting from scratch. Also, commands are available in Microsoft Word that will allow a Word document containing MathType (or Equation Editor) equations to be converted to TEX or any other language supported by a translator. MathType's translation facilities can be used as an aid to learning TEX, as a simpler interface for entering equations into a TEX authoring system, or as part of a document conversion scheme for journal and book publishers.

The MTCode character encoding

Although the designers of Unicode have attempted to include many mathematical characters, their attempt falls somewhat short. In fairness to them, incorporating all the characters of the many natural languages in use in the world must have been an overwhelming task.

There is an attempt by some in the mathematical community to get the Unicode Consortium to add the missing mathematical characters to a future version of Unicode. If and when they are successful, we will probably adopt it to replace MTCode.

MathType uses each character's MTCode value as a key into a database of character information that, for each character, includes a human-readable description, an indicator of its role in mathematical notation (e.g. variable, binary operator), and information used in the process of choosing an appropriate font to render it on screen and printer. Most importantly, a character's MTCode value is an index into tables of translation strings in MathType's translation system.

We will use the terms MTCode and Unicode interchangeably in the remainder of this paper.

[1] Unicode is a standard for encoding characters. See www.unicode.org for information.

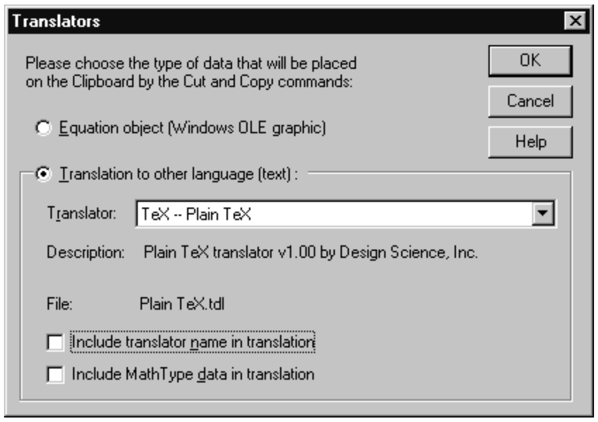
The translation system from a user's perspective

The basic scenario for using MathType to aid in the creation of a TeX document is to run it simultaneously with your favorite TeX editor. The process is this:

- whenever an equation is needed, the MathType window is brought to the front;
- the equation is created in the MathType window;
- the equation is selected and copied to the clipboard, a process which invokes the previously selected translator;
- the TeX editor is brought to the front;
- the TeX code for the equation is pasted into the document.

Editing to correct mistakes is performed by reversing this process, pasting the TeX code (along with a comment containing a compressed form of MathType's internal representation) back into a MathType window, and then repeating the above process once the corrections have been made.

At the beginning of such a document creation and editing session, the user must select one of MathType's translators. This is done via the Translators dialog, which presents a list of all the translators present on the user's system in the MathType translators directory (Fig. 2).


Translators

Please choose the type of data that will be placed on the Clipboard by the Cut and Copy commands:

( ) Equation object (Windows OLE graphic)

(•) Translation to other language (text) :

Translator: TeX -- Plain TeX

Description: Plain TeX translator v1.00 by Design Science, Inc.

File: Plain TeX.tdl

[ ] Include translator name in translation

[ ] Include MathType data in translation

Figure 2: The Translators dialog

Anatomy of a translator

Each translator is defined by a text file written in a simple language called TDL (Translator Definition Language). A translator has a simple structure:

- The first line defines the short name for the translator which appears in the list presented to the user in the Translators dialog (see Fig. 2) and a longer description which appears in the dialog once the translator is chosen from the list. The description might include the author's name and affiliation as well as the version number of the translator.
- a set of matching rules of the form
  ⟨*thing to match*⟩ = ⟨*translation string*⟩ ;

MathType equations (just like TeX ones) are represented internally as a tree. Let's take the following equation as an example:

$$y = \frac{a+b}{c}$$

MathType sees this equation as:

```
eqn (root)
  slot (main)
    character (y)
    character (=)
    template (fraction)
      slot (numerator)
        character (a)
        character (+)
        character (b)
      slot (denominator)
        character (c)
```

The translation process begins by applying the display equation rule,[2] to the root of the MathType expression:

```
     eqn = "\[@n#@n\]@n";   // display equation
```

The characters between quotation marks are processed from left to right. Most of the characters in the `eqn` rule are simply placed in the output translation stream. The `@n` sequence outputs a newline. The `@` character, called the escape character, is used to insert special characters into the output stream. The default escape character is `$`, but is redefined in the TeX translators to `@` for convenience.

The `#` in the `eqn` rule causes the translator to look for a rule that will be used to translate the equation's main slot. After applying this translation (we'll get to that next) and inserting its output into the translation stream, the rest of the `eqn` translation string is output and the translation process is complete.

Let's go back to see how the `#` in the `eqn` rule is processed. This is done with the rule:

```
slot/t = "#";
```

This rule works just like the `eqn` rule but is even simpler. The `/t` option is used to signal that this

[2] The translation rules used in our example are simplified somewhat for the purpose of this paper.

rule is to be used for the top-most slot in the equation only. Other slot rules enclose the translation of their contents in {}, the TEX notation for grouping. Now, each item in the main slot is processed. The rules for y and = are also very simple:

```
char/0x0079 = "y";    // Latin small letter y
char/0x003D = " = ";  // Equals sign
```

Here is where Unicode comes into play. MathType knows the Unicode value for y is 79 (in hexadecimal notation). It uses this knowledge to find the char rule that specifies how y is to be translated.

The fraction is handled by a template rule:

```
frac = "\frac{#1}{#2}";  // fraction
```

The #s in this rule are followed by numerals that specify the slot's index in the template; 1 for the numerator, 2 for the denominator. These two slots are processed much like the main slot, except they use the more general slot rule:

```
slot = "{#}";
```

So, the complete translation of our simple example is:

```
\[
y = \frac{a+b}{c}
\]
```

MathML

The MathML specification was written by the W3C Math Working Group.[3] In April 1998, it was raised to Recommendation status by the W3C. MathML has as its main goals:

- encode mathematical material suitable for teaching and scientific communication at all levels
- encode both mathematical notation and mathematical meaning

MathML is intended to be used to both present mathematical notation and to serve as as a medium of exchange between scientific and mathematical software. Toward that end, MathML defines a set of XML elements and attributes (together called markup) that fall into two categories: presentation markup and content markup. Presentation markup is intended to describe mathematical expressions from a two-dimensional layout point of view, whereas content markup is intended to capture the meaning of the mathematics.

MathType provides four MathML translators (why there are four will be explained shortly) that can convert its equations into MathML's presentation markup. For example, translating the following expression into MathML:

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

results in:

```
<math displaystyle='true'>
   <mrow>
      <mfrac>
         <mrow>
            <mo>-</mo>
            <mi>b</mi><mo>&PlusMinus;</mo>
            <msqrt>
               <mrow>
                  <msup>
                     <mi>b</mi>
                     <mn>2</mn>
                  </msup>
                  <mo>-</mo>
                  <mn>4</mn><mi>a</mi><mi>c</mi>
               </mrow>
            </msqrt>
         </mrow>
         <mrow>
            <mn>2</mn><mi>a</mi>
         </mrow>
      </mfrac>
   </mrow>
</math>
```

The first reaction of most TEX users is a gasp at how verbose MathML is. Please bear in mind that MathML is not intended to be authored directly by humans but with tools like MathType. MathML inherits its verbosity from XML. This "disadvantage" is far outweighed by the advantages gained with XML structure with its support in browsers, editors, and other tools. In the future, we hope to see MathML become the language of choice for exchanging mathematics between mathematical applications. Its eventual integration with browsers should make it tremendously useful in teaching.

Unfortunately, it may be a little while before MathML achieves its promise. Until we are able to properly display MathML in the popular browsers, such as Microsoft's Internet Explorer and Netscape's Navigator, we will have to rely on various browser "plug-ins", such as IBM's techexplorer.[4] and Geometry Technologies' WebEQ.[5] These work but are constrained by various font issues, sizing problems, and lack baseline alignment for in-line math. The W3C's

[3] See http://www.w3.org/Math/ for the specification and other information on MathML.

[4] For information, see http://www.software.ibm.com/network/techexplorer/.

[5] See http://www.webeq.com/.

Math Working Group has made browser integration one of its highest priorities.

In order to work with the various MathML browser plug-ins, we have had to create several versions of our MathML translator, one for each. They differ only in the “wrapper” code they place around the generated MathML in order to invoke the plug-in and pass the MathML code to it. When true browser integration is possible, we will only need a single MathML translator.

Experimenting with MathType’s translators

Because of their simple and open structure, MathType’s translators can be easily modified or new ones written from scratch. Some possibilities include:

- creating a new translator for the mathematics portion of an SGML document standard (DTD)
- changing an existing TeX translator to make use of some macro package or the author’s own preferred macros
- using the Unicode capability of MathType’s translators to take advantage of the various TeX adaptations for non-Ennglish languages

Document conversion

MathType’s translation facilities can also be incorporated into the process of converting entire documents to TeX or any other language supported by one of its translators. MathType has a programmatic interface beside its more familiar graphical user interface. This interface is via functions in a Windows DLL (Dynamically Linked Library). As MathType is often used with Microsoft Word, we have provided functionality that can be accessed using commands on a “MathType” menu within the Word application itself. One of these is a Convert Equations command that can be used to convert all the MathType and Equation Editor equations in a document to any one of the languages for which a MathType translator is available. Although this does not result in a fully translated Word document, the most difficult part, converting equations, has been achieved.

MathType’s Word support is written in that product’s Visual Basic language. The source code is accessible and may be used as the basis for your own conversion scheme.

Conclusions

MathType 4.0, with its new translation features, can be used in a variety of ways:

- as an interactive front-end to TeX authoring
- as an aid to learning TeX
- to experiment with the new MathML standard
- with the development of custom translators, it can be used to generate virtually any text-based math language

It is our hope that TeX users will want to add it to their arsenal of useful tools.

TeX musings

Musings from the Bard

Oh, what a tangled web is TeX,
or so it seems at the outset;
for highest quality, the best to look,
Oh why did I choose to typeset my own book!
...
For Macintosh users the skies are quite blue,
with Barry to help you and Ben Salzburg too.
With Art, Ross, and Uwe ready to assist,
just send a short email to Gary Gray’s (Textures) list.
...
If shareware type software is more to your taste,
your super-fast Power-Mac need not go to waste.
There’s CMac- and Direct-TeX to lessen the sorrow,
and that great program OzTeX, by Andrew Trevorrow.
...
For Unix-like platforms the software’s all free,
with a teTeX installation from the TeX-Live CD,
which collects all the pieces and orders all parts,
Thanks to Thomas Esser, Kaja, and Sebastian Rahtz.
...
Leslie Lamport created LaTeX nearly fifteen years ago.
It evolved into 2-epsilon by a process rather slow.
Improvements are numerous; results you can see.
Thanks to Carlisle, Mittelbach and Rowley,
It’ll be even better with release LaTeX3.
...
For PostScript Type1 fonts exquisitely drawn,
The expert is Berthold, whose surname is Horn.
Where the yin meets the yang in the great cosmic goo,
Don’t get this name mixed-up with Louis Vosloo.

—Ross Moore