\blah{TEX}

blahtex version 0.4 manual

David Harvey

1 Introduction

This is the manual for blahtex version 0.4. The most up-to-date information about blahtex is hopefully available at

`www.blahtex.org`

1.1 How this document is organised

- **What blahtex can handle** (Section 2) explains what kind of TeX input blahtex can cope with, and how it differs from texvc.
- **The blahtex command-line application** (Section 3) describes how to compile, install, and run the blahtex command-line application, and how to interpret its output. This will be of interest to developers who would like a simple way to incorporate blahtex into their project.
- **The blahtex API** (Section 4) describes how to link blahtex directly into your code, which might give better performance in some environments.

1.2 What is blahtex?

Blahtex is a free software tool/library that translates TeX markup into MathML markup. It is also capable of generating PNG format images, using some external tools.

Blahtex is *not* designed to process entire TeX documents. Rather, it focuses on the mathematical capabilities of the TeX language, processing only a single equation at a time. It is designed to provide mathematical support to a larger

document markup system. Currently, the main target platform is MediaWiki — the software that powers Wikipedia and many other wikis — but blahtex has been designed with flexibility of integration in mind.

Blahtex concentrates on matching the *appearance* of TeX output, as far as this is possible given the fonts available to the MathML renderer. It only outputs Presentation MathML, not Content MathML. Blahtex is aware of at least some of TeX's rules concerning spacing and fonts. For example, it knows about "atom flavours" (like ord, rel, op, etc) and TeX's algorithms for determining the amount of space between them.

Blahtex implements some subset of TeX, LaTeX and AMS-LaTeX. A complete list of supported and quasi-supported commands can be found in Section 2. More complete coverage is planned for future versions.

Blahtex is internally Unicode-based. Non-ASCII characters, such as extended Latin alphabets or even Chinese characters, may be used in text mode (e.g. within `\text{...}` blocks). These will be handled correctly for MathML output. For PNG output, only the simplest characters handled by the LaTeX `ucs` package can presently be used; a complete list is shown in Section 2.20.

Blahtex is open source software. The source code is released under the GNU GPL (General Public License). This means that although the source is copyrighted, you may modify it, use it in your own programs, or even sell it, as long as you adhere to the GPL. It is written in C++, and has been completely rewritten since the last public release (blahtex 0.2.1).

Blahtex obviously owes a lot to texvc, the software presently used by MediaWiki to handle TeX input, written by Tomasz Wegrzanowski.

Blahtex is a work in progress. I hereby solicit **your feedback**, to help me improve it as much as possible.

(It has not escaped the author's attention that every paragraph of this section either begins or ends with the word 'blahtex'.)

1.3 The origin of the name 'blahtex'

In the beginning there was TeX. Later, we also met LaTeX, and ConTeXt, teTeX, MiKTeX, blah blah blah...

1.4 Other converters

There are a variety of other TeX-to-MathML converters available. The MathML home page (`http://www.w3.org/Math/`) has quite a long list. Here are a few that have online demos available:

- **itex2mml**:
  `http://pear.math.pitt.edu/mathzilla/itex2mml.html`
- **TexToMathML**:
  `http://www.orcca.on.ca/MathML/texmml/textomml.html`

- **TtM**:
  `http://hutchinson.belmont.ma.us/tth/mml/`

They have their pros and cons, as does blahtex. I happen to think blahtex is rather good, but of course I am biased :-) Feel free to disagree. Please let me know if you think blahtex is no good, and *why* it's no good, so that maybe I can fix it. (Also, let me know if you think it's great!)

1.5 Acknowledgements

Thanks to the crew at Wikipedia, for pioneering such a fabulous resource, and especially the regulars at WikiProject Mathematics.

Thanks to Jitse Niesen for his ongoing work on integrating blahtex into MediaWiki, and for generally being very supportive of this project.

2 What blahtex can handle

Blahtex supports some subset of TeX, LaTeX and AMS-LaTeX. This section gives a complete list of supported commands, together with some comments where the support is known to be incomplete.

2.1 Newcommand

Blahtex supports `\newcommand`, including arguments (but not *optional* arguments).

Blahtex protects against a malicious user eliciting exponential time via recursive macros, by imposing a hard limit on the amount of macro processing that can occur.

Note that `\newcommand` is *not* local to blocks, as is the case in TeX. For example, {`\newcommand{\abc}{xyz}`} `\abc` is legal in blahtex, but not in TeX, because the definition of `\abc` is only remembered within the outermost {...} block.

Clearly `\newcommand` is not very useful for an individual equation. In a larger document markup system, a good approach might be to provide a facility for specifying a document-wide collection of macros, and the software would automatically append the relevant `\newcommand`s to the beginning of each equation in which a macro need to be available. It is not clear at this stage whether this model would be technically feasible in MediaWiki.

2.2 Environments

`\begin{XYZ}` ... `\end{XYZ}`, where `XYZ` is one of:

```
matrix pmatrix bmatrix Bmatrix vmatrix Vmatrix
cases aligned smallmatrix
```

There are still some spacing issues with the `aligned` environment.

2.3 Miscellaneous

\sqrt (including with optional argument)
\substack (vertical spacing is not quite right yet)
\overset \underset \not

Support for \not is not entirely correct. Simple things like \not= will work fine. It is *not* implemented as a character that gets drawn over the previous character, as is done in TeX. Rather, it is a command that takes a single argument, and selects an appropriate MathML character representing the negation of the argument. The list of characters supported is still rather thin.

2.4 Text commands

\text \textit \textbf \textrm \texttt \textsf \emph \hbox \mbox

2.5 Fractions, binomials

\frac \cfrac \over \binom \choose \atop

2.6 Delimiters

\left \right
\big \Big \bigg \Bigg
\bigl \Bigl \biggl \Biggl
\bigr \Bigr \biggr \Biggr

2.7 Atom flavours

\mathop \mathrel \mathord \mathbin \mathopen \mathclose \mathpunct \mathinner

2.8 Limits

\limits \nolimits \displaylimits

2.9 Spacing

\, \! \ \; \> \quad \qquad

2.10 Accents

\hat \widehat \dot \ddot \bar \overline \underline \overbrace \underbrace \overleftarrow \overrightarrow \overleftrightarrow \check \acute \grave \vec \breve \tilde \widetilde

2.11 Fonts

\mathbf \mathbb \mathrm \mathit \mathcal \mathfrak \mathsf \mathtt
\boldsymbol \rm \bf \it \cal \tt \sf \Bbb \bold

2.12 Style

\displaystyle \textstyle \scriptstyle \scriptscriptstyle

2.13 Named operators

\operatorname \operatornamewithlimits \lim \sup \inf \limsup \liminf
\injlim \projlim \varlimsup \varliminf \varinjlim \varprojlim \min
\max \gcd \det \Pr \ker \hom \dim \arg \sin \cos \sec \csc \tan \cot
\arcsin \arccos \arctan \sinh \cosh \tanh \coth \log \lg \ln \exp \deg
\mod \bmod \pmod

2.14 Escaped characters

_ \& \$ \# \% \{ \}

2.15 Greek letters

\alpha \beta \gamma \delta \epsilon \varepsilon \zeta \eta \vartheta
\theta \iota \kappa \varkappa \lambda \mu \nu \pi \varpi \rho \varrho
\sigma \varsigma \tau \upsilon \phi \varphi \chi \psi \omega \xi
\digamma \Gamma \Delta \Theta \Lambda \Pi \Sigma \Upsilon \Phi \Psi
\Omega \Xi

2.16 Various mathematical symbols

\implies \neg \ne \ge \le \land \lor \gets \to \vert \lvert \rvert
\Vert \lVert \rVert \lfloor \rfloor \lceil \rceil \lbrace \rbrace
\langle \rangle \lbrack \rbrack \aleph \beth \gimel \daleth \wp \ell
\P \imath \forall \exists \Finv \Game \partial \Re \Im \leftarrow
\rightarrow \longleftarrow \longrightarrow \Leftarrow \Rightarrow
\Longleftarrow \Longrightarrow \mapsto \longmapsto \leftrightarrow
\Leftrightarrow \longleftrightarrow \Longleftrightarrow \uparrow
\Uparrow \downarrow \Downarrow \updownarrow \Updownarrow \searrow
\nearrow \swarrow \nwarrow \hookrightarrow \hookleftarrow
\upharpoonright \upharpoonleft \downharpoonright \downharpoonleft
\rightharpoonup \rightharpoondown \leftharpoonup \leftharpoondown
\nleftarrow \nrightarrow \supset \subset \supseteq \subseteq
\sqsupset \sqsubset \sqsupseteq \sqsubseteq \supsetneq \subsetneq
\in \ni \notin \iff \mid \sim \simeq \approx \propto \equiv \cong \neq
\ll \gg \geq \leq \triangleleft \triangleright \trianglelefteq
\trianglerighteq \models \vdash \Vdash \vDash \lesssim \nless \ngeq

\nleq \times \div \wedge \vee \oplus \otimes \cap \cup \sqcap \sqcup \smile \frown \smallsmile \smallfrown \setminus \smallsetminus \And \star \triangle \wr \infty \circ \hbar \lnot \nabla \prime \backslash \pm \mp \emptyset \varnothing \S \angle \colon \Diamond \nmid \square \Box \checkmark \complement \eth \hslash \mho \flat \sharp \natural \bullet \dagger \ddagger \clubsuit \spadesuit \heartsuit \diamondsuit \top \bot \perp \ldots \cdot \cdots \vdots \ddots \dots \dotsb

2.17 Large operators

\sum \prod \int \iint \iiint \iiiint \oint \bigcap \bigodot \bigcup \bigotimes \coprod \bigsqcup \bigoplus \bigvee \biguplus \bigwedge

2.18 Symbols only available in text mode

\O \" \' \textbackslash \textvisiblespace \textasciicircum \textasciitilde

2.19 Special commands

If the magic command \strictspacing occurs anywhere in the input, blahtex will switch to 'strict spacing mode' for the entire equation. This overrides the command-line --spacing setting.

2.20 Non-ASCII characters

Only the following non-ASCII characters may be used when generating PNG output (assuming also that the --use-ucs-package command-line option is enabled):

¡ £ § © ¬ ® ° μ ¶ ¿ À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý ß à á â ã ä å æ ç è é ê ë ì í î ñ ò ó ô õ ö ÷ ø ù ú û ü ý ÿ Ā ā Ă ă Ć ć Ĉ ĉ Ċ ċ Č č Ď ď Ē ē Ĕ ĕ Ė ė Ě ě Ĝ ĝ Ğ ğ Ġ ġ Ģ Ĥ ĥ Ĩ ĩ Ī ī Ĭ ĭ İ ı Ĵ ĵ Ķ ķ Ĺ ĺ Ļ ļ Ľ ľ Ł ł Ń ń Ņ ņ Ň ň Ō ō Ŏ ŏ Ő ő Œ œ Ŕ ŕ Ŗ ŗ Ř ř Ś ś Ŝ ŝ Ş ş Š š Ţ ţ Ť ť Ũ ũ Ū ū Ŭ ŭ Ů ů Ű ű Ŵ ŵ Ŷ ŷ Ÿ Ź ź Ż ż Ž ž Ǎ ǎ Ǐ ǐ Ǒ ǒ Ǔ ǔ Æ æ Ǧ ǧ Ǩ ǩ ǰ Ǵ ǵ Ǹ ǹ Ǽ ǽ Ǿ ǿ Ș ș Ț ț Ȟ ȟ Ȧ ȧ Ę ę Ȯ ȯ Ȳ ȳ

Currently blahtex does not recognise TeX's accent commands (like \"o), so it is necessary to enter such characters in Unicode directly.

2.21 Differences between blahtex and texvc

2.21.1 HTML support

The main feature of texvc that is missing in blahtex is support for HTML output. This may or may not be added in future.

2.21.2 Error reporting

Blahtex has much more robust syntax error reporting than texvc. Rather than a handful of generic error messages, blahtex can generate a wide variety of more detailed error messages to help the user diagnose the problem.

2.21.3 Parsing differences

Blahtex generally achieves much higher compatibility with TeX's parsing than texvc does. Texvc is generally more permissive. For example, the following are legal in texvc, but in TeX and blahtex they require additional grouping braces:

- `\frac \sqrt a \hat b`
- `x^\cong`
- `x^\left( xyz \right)`
- `x^\begin{matrix} a \end{matrix}`

The characters `$` and `%` are legal in texvc, but are illegal in blahtex. (Of course `\$` and `\%` are available.)

These parsing differences may cause problems in replacing texvc with blahtex in an existing MediaWiki installation, since some legacy equations may not be compatible with blahtex. Preliminary research suggests that about 0.5% of equations on Wikipedia itself (including the ten largest language Wikipedias) would be affected.

2.21.4 Nonstandard commands

Blahtex has a command-line option (`--texvc-compatible-commands`) that enables all of the nonstandard commands in texvc's dialect of TeX; that is, commands which are not present in TeX, LaTeX, or AMS-LaTeX. It appears that most of these commands were added to texvc to make life easier for people familiar with HTML entities; for example, `\isin` is a texvc synonym for the standard `\in`. This option should be useful for backward compatibility with existing equations in databases like Wikipedia. Here is the complete list:

```
\R \Reals \reals \Z \N \natnums \Complex \cnums \alefsym \alef \larr
\rarr \Larr \lArr \Rarr \rArr \uarr \uArr \Uarr \darr \dArr \Darr
\lrarr \harr \Lrarr \Harr \lrArr \hAar \sub \supe \sube \infin \lang
\rang \real \image \bull \weierp \isin \plusmn \Dagger \exist \sect
\clubs \spades \hearts \diamonds \sdot \ang \thetasym \Alpha \Beta
\Epsilon \Zeta \Eta \Iota \Kappa \Mu \Nu \Rho \Tau \Chi \arcsec
\arccsc                          \arccot                          \sgn
```

Also included are the four commands `\empty`, `\and`, `\or`, `\part`. These commands *are* part of TeX/LaTeX/AMS-LaTeX, but they do *not* do what texvc thinks they should do! Blahtex emulates texvc's behaviour for these commands.

2.21.5 Additional commands

Commands supported by blahtex but not by texvc include:

`\newcommand` `\substack` `\overset` `\underset` `\text` `\bigl` `\Bigl` `\biggl` `\Biggl` `\bigr` `\Bigr` `\biggr` `\Biggr` `\mathrel` `\mathord` `\mathbin` `\mathopen` `\mathclose` `\mathpunct` `\mathinner` `\widetilde` `\displaystyle` `\textstyle` `\scriptstyle` `\scriptscriptstyle` `\operatornamewithlimits` `\textbackslash` `\textasciicircum` `\textasciitilde` `\begin{aligned}` `\begin{smallmatrix}`

Also, blahtex improves support for `\binom`, `\mathop`, and `\limits`, which do not always work correctly in texvc.

3 The blahtex command-line application

The blahtex source code is available from `www.blahtex.org`. No binaries will be made available. All official releases should have been signed with a PGP key whose ID is 0x6269E206 and whose fingerprint is

```
9A51 0B6A B144 6A4D E1E5 0DE6 D604 6405 6269 E206
```

This key is valid until 2nd August 2007. You can either get it from the blahtex website, or try searching for "blahtex" on any public keyserver.

Besides reading this document, the interested developer is strongly advised to "use the source".

3.1 Compatible systems

Blahtex has been successfully compiled and run on the following configurations:

- Linux with gcc 4.0.2 20050808 (prerelease)
- Mac OS 10.3.9 with gcc 3.3
- Mac OS 10.4.3 with gcc 4.0.1

You will probably encounter problems with compilers other than gcc, or with older versions of gcc. I know of at least one older Solaris compiler that couldn't stomach the code. If you want to compile it on MS Windows — good luck.

To generate PNGs, you will need LaTeX and ImageMagick installed. Blahtex has been successfully tested with ImageMagick version 6.2.4. If you use a sufficiently old version of ImageMagick (perhaps 5.x), the transparent PNGs will break. If it is not possible for you to upgrade an old ImageMagick installation, you might consider the blahtex command-line option `--convert-options "-quality 100 -density 120"`, which will disable support for transparent PNGs.

3.2 Compiling blahtex

Unpack the source into your favourite directory. Change directory to wherever the makefile is, and then:

- If you're running Linux, just type `make linux`.
- If you're running Mac OS X (as I do), try `make mac`.

You should then find an executable `blahtex` in the current directory. If you want to quickly test it, try `echo '\frac xy' | ./blahtex --mathml`.

Note that you will need the GNU `iconv` library installed before you can compile. On some systems this is preinstalled, so you don't need to do anything. On my Mac I needed to install it (for example via fink).

Some of the source files seem to need a bit of memory to compile. I had trouble with `-O3` level optimisation on an older machine with 256MB RAM and gcc 3.3. It should be fine with 512MB or above.

3.3 Command-line syntax

The basic syntax is: `blahtex [ options ]`; the command-line options are listed below. The TeX input should be supplied on standard input in UTF-8 encoding, which means plain ASCII if you don't care about Unicode. If no input is given, blahtex will print a help screen. If neither of the `--mathml` or `--png` options are selected, then blahtex will still process the input for syntax errors, but will product no output.

3.3.1 General options

- `--help`. Prints out a list of command-line options.
- `--texvc-compatible-commands`. Enables use of commands that are specific to texvc, but that are not standard TeX/LaTeX/AMS-LaTeX commands (see section 2.21.4).
- `--throw-logic-error`. Simulates the effect of a debug assertion occurring, so that you can test any associated error-logging code.
- `--debug` *`type`*. Enables some debugging output to assist in working out what is going on inside blahtex's head:
  - `--debug parse`. Print the parse tree.
  - `--debug layout`. Print the layout tree. This is an intermediate stage between parsing and MathML.
  - `--debug purified`. Print 'purified TeX'. This is the complete TeX file that blahtex sends to LaTeX for PNG generation.

  Multiple `--debug` options may be present. The format of debugging output is subject to change, and is not designed to be machine-readable; it will interrupt blahtex's usual XML output format in ghastly ways.

- `--print-error-messages`. This will print out a list of all error IDs and corresponding messages (on alternating lines) that blahtex can possibly emit inside an `<error>` block (see Section 3.4).

3.3.2 MathML-related options

- `--mathml`. Enables MathML output.
- `--mathml-encoding` *`type`*. Controls the way blahtex outputs MathML characters.
  - `--mathml-encoding raw`. Use Unicode code points (i.e. UTF-8) directly in the output.
  - `--mathml-encoding numeric` (default). Use XML numeric entities, like `&#x2191;`. This is likely to be the most portable option.
  - `--mathml-encoding short`. Use 'short' MathML entity names, like `&uarr;`.
  - `--mathml-encoding long`. Use 'long' MathML entity names, like `&UpArrow;`.

  Not every MathML character has 'short' and/or 'long' names; blahtex will fall back on numeric entities in this case.
- `--disallow-plane-1`. Prevents blahtex from outputting any plane-1 Unicode characters, either as UTF-8 or as numeric entities. Instead, it will use named entities like `&Afr;` (Fraktur 'A'). The rationale is that some browsers have somewhat incomplete support for plane-1 characters, but do okay with these named entities.
- `--mathml-version-1-fonts`. Forbids use of the `mathvariant` attribute, which is only available in MathML 2.0. Instead, blahtex will use MathML version 1.x font attributes: `fontfamily`, `fontstyle` and `fontweight`, which are all deprecated in MathML 2.0. If these attributes are insufficient, for example characters with `mathvariant` equal to `double-struck`, blahtex will substitute explicit MathML entities.
- `--other-encoding` *`type`*. Controls the way blahtex outputs non-ASCII, non-MathML characters. Such a character could only occur if it was supplied directly in the input.
  - `--other-encoding raw`. Use Unicode code points (i.e. UTF-8) directly in the output.
  - `--other-encoding numeric` (default). Use XML numeric entities.

  Note: the default values for `--mathml-encoding` and `--other-encoding` imply that all output is plain ASCII.

- `--indented`. Prints each MathML tag on a separate line, with appropriate indenting.
- `--spacing` *`type`*. Controls how much MathML spacing markup to use (i.e. `<mspace>` tags, and `lspace`/`rspace` attributes). Blahtex always uses TeX's rules (or an approximation thereof) to compute how much space to place between symbols in the equation, but this option describes how often it will actually emit MathML spacing markup to implement its spacing decisions.
  - `--spacing strict`. Output spacing markup everywhere possible; leave as little choice as possible to the MathML renderer. This will result in the most bloated output, but hopefully will look as much like TeX output as possible.
  - `--spacing moderate` (default). Output spacing commands whenever blahtex thinks a typical MathML renderer is likely to do something visually unsatisfactory without additional help. The aim is to get good agreement with TeX without overly bloated MathML markup. (It's very difficult to get this right, so I expect it to be under continual review.)
  - `--spacing relaxed`. Only output spacing commands when the user specifically asks for them, using TeX commands like `\,` or `\quad`.

  The magic command `\strictspacing` will override this setting (see Section 2.19).

  Blahtex pays a lot of attention to spacing, because the MathML defaults (via the operator dictionary) are often insufficient. To see the difference, try the simple input `a := b` on blahtex (with spacing set to moderate or strict) and compare with the output of other translators.

3.3.3 PNG-related options

- `--png`. Enables PNG output. For this to work, you need to have latex, dvips, and ImageMagick installed.
- `--use-ucs-package`. This tells blahtex it may use the LaTeX `ucs` package to handle certain non-ASCII characters (see Section 2.20 for a list). Obviously, it is necessary to install the `ucs` package before using this option.
- `--shell-latex` *`command`*. Specifies the command to use for running LaTeX. Default is just `latex`.
- `--shell-dvips` *`command`*. Specifies the command to use for running dvips. Default is just `dvips`.
- `--shell-convert` *`command`*. Specifies the command to use for running ImageMagick. Default is just `convert`.

- `--convert-options` *`options`*. Specifies the command-line options to pass to `convert`. The blahtex default is `-quality 100 -density 120 -matte -fill None -opaque White`. This is the same as texvc's setting, except that it also incorporates a suggestion of Maynard Handley that makes the PNG transparent.

- `--temp-directory` *`directory`*. Specifies the directory that should be used for the intermediate files used during PNG creation. Default is the current directory.

- `--png-directory` *`directory`*. Specifies the directory in which the PNG output file should be placed. Default is the current directory.

3.4 Interpreting blahtex's output

Blahtex's output looks like XML. (Unless a *really fatal* error occurs :-)) By default, the output is completely ASCII, although there are command-line options which enable UTF-8 output for certain characters. The entire output is surrounded by the tags `<blahtex>...</blahtex>`. Inside these tags, there are several possibilities:

- If a debug assertion occurred (i.e. if blahtex detected a bug within itself), you will see a `<logicError>...</logicError>` block. Between the `<logicError>` tags will be a string describing the error. If you ever see one of these, please report it to me.

- If there was a syntax error in the TeX input, there will be a single `<error>...</error>` block which describes the error (the `<error>` block format is described in detail below). The possible error IDs that can occur here are:

NonAsciiInMathMode	IllegalCharacter
ReservedCommand	TooManyTokens
IllegalFinalBackslash	UnrecognisedCommand
IllegalCommandInMathMode	IllegalCommandInMathModeWithHint
IllegalCommandInTextMode	IllegalCommandInTextModeWithHint
MissingOpenBraceBefore	MissingOpenBraceAfter
MissingOpenBraceAtEnd	NotEnoughArguments
MissingCommandAfterNewcommand	IllegalRedefinition
MissingOrIllegalParameterCount	MissingOrIllegalParameterIndex
UnmatchedOpenBracket	UnmatchedOpenBrace
UnmatchedCloseBrace	UnmatchedLeft
UnmatchedRight	UnmatchedBegin
UnmatchedEnd	UnexpectedNextCell
UnexpectedNextRow	MismatchedBeginAndEnd
CasesRowTooBig	SubstackRowTooBig
InvalidNegation	MissingDelimiter
IllegalDelimiter	MisplacedLimits
DoubleSuperscript	DoubleSubscript
AmbiguousInfix	UnavailableSymbolFontCombination
InvalidUtf8Input	

- Assuming there were no syntax errors or debug assertions:
  - If you gave the `--mathml` option at the command line, you will get a `<mathml>...</mathml>` block. If the MathML was generated successfully, the `<mathml>` block will contain a `<markup>...</markup>` block, containing the actual MathML. If there was a problem generating the MathML, the `<mathml>` block will instead contain an `<error>` block describing the problem. The only possible error ID here is `TooManyMathmlNodes`.
  - If you gave the `--png` option at the command line, you will get a `<png>...</png>` block. If the PNG image was generated successfully, then it will be stored in a file called `X.png`, where `X` is an md5 hash (32 character lowercase hex string); the `<png>` block will then contain `<md5>X</md5>`. (In fact `X` is the md5 hash of the TeX file that got sent to LaTeX to generate the image.) If there was an error generating the PNG file, the `<png>` block will instead contain an `<error>` block describing the problem. The possible error IDs here are:

PngIncompatibleCharacter	CannotCreateTexFile
CannotWriteTexFile	CannotRunLatex
CannotRunDvips	CannotRunConvert
CannotChangeDirectory	

The `<error>` block (mentioned several times above) has the following format. First, it contains an `<id>...</id>` block, containing an error ID (i.e. one of the CamelCase strings listed above). Next, a sequence of zero or more `<arg>...</arg>` blocks, representing the 'arguments' of the error. Finally there

is a `<message>...</message>` block, containing a translation of the error into English. For example, one possible error block is:

```
<error>
<id>MismatchedBeginAndEnd</id>
<arg>\begin{matrix}</arg>
<arg>\end{array}</arg>
<message>The commands "\begin{matrix}" and
"\end{array}" do not match</message>
</error>
```

The simplest way to report the error to the user is to extract the `<message>` block. If you want to implement some localisation of error messages, you should use the `<id>` and `<arg>` fields. A complete list of error messages can be found in the source file `Messages.cpp`, or try the command-line option `--print-error-messages`. The error IDs may change in future versions of blahtex.

4 The blahtex API

This section gives a summary of how to link blahtex directly into a C++ application. You will need to write a wrapper if you want to use a different language. (If you do this, please consider sending me the wrapper so I can make it available for others to use.)

4.1 Core vs non-core

The blahtex source code is divided into two parts:

- The 'blahtex core', whose source files are all in the `BlahtexCore` subdirectory. The core does all the hard work involved in translating TeX to MathML, and the not-as-hard work of preparing a complete TeX file to be sent to LaTeX to generate the PNG image. It does not include any functionality which may be more OS-dependent; pretty much all it does is allocate memory and push strings around.
- The blahtex command-line application, whose source files are in the main `source` directory. This 'non-core' source is basically a wrapper that turns the blahtex core into a command-line application, and additionally handles shelling out to LaTeX to generate the PNG output.

4.2 How to use the core

To use the blahtex core in your C++ application, you should follow these steps:

1. Copy the `BlahtexCore` directory to wherever your project is.

2. Any source file that wants to access the blahtex core needs to `#include "BlahtexCore/Interface.h"`.

3. Everything in the blahtex core is in the `blahtex` namespace. So, you might also consider `using namespace blahtex`.

4. Declare an object of type `blahtex::Interface`. (It's perfectly okay to have several `Interface` objects lying around; they won't get in each other's way.)

5. You can set various conversion options by setting the public member variables of the `Interface` object. See the header file `Interface.h` for a list of members. The structs `MathmlOptions`, `EncodingOptions` and `PurifiedTexOptions` are described in detail in the header file `Misc.h`; they basically correspond to various command-line options (see Section 3.3).

6. Call the member function `Interface::ProcessInput(x)`, where `x` is a `wstring` containing the input TeX.

7. You can call the member function `Interface::GetMathml()` to get the MathML translation as a `wstring`.

8. You can call the member function `Interface::GetPurifiedTex()` to get the 'purified TeX' as a `wstring`; this is a complete TeX file that could be sent to LaTeX to generate graphical output.

9. Any of the above functions can throw exception objects if something goes wrong, so you probably need to worry about `catch`ing them. They will throw a `std::logic_error` object if a debug assertion occurs. They will throw a `blahtex::Exception` object to indicate a syntax error in the input, or if there is a problem in generating the MathML or purified TeX. The `blahtex::Exception` object is documented in `Misc.h`. If you need the error translated to English, you probably want to check out the `GetErrorMessage` function in `Messages.cpp` (not part of the blahtex core).

4.3 Dealing with `wstring`

The blahtex core is internally Unicode throughout, and works exclusively with wide strings — `wstring`, not `string`. If your code only deals with ASCII strings, or UTF-8, you will need a way of converting between narrow and wide strings. The blahtex command-line application has a class `UnicodeConverter` which provides precisely this functionality; it is essentially a C++ wrapper for the `iconv` library in terms of `string` (for storing UTF-8 strings) and `wstring` (for storing UCS-32 strings; endianness depends on the platform). To use this class:

1. Put `UnicodeConverter.cpp` and `UnicodeConverter.h` in your project directory, and make sure you `#include "UnicodeConverter.h"`.

2. Link against the `iconv` library. You may need to compile and install `iconv`, and possibly use the linker switch `-liconv`.

3. On some systems (including Mac OS X, but not Linux), you need to define the constant `BLAHTEX_ICONV_CONST` for `UnicodeConverter.cpp`, otherwise you'll probably get compiler warnings. See the source for an explanation.

4. Declare a `UnicodeConverter` object and call `Open()`. This sets up the underlying `iconv_t` handles.

5. Use the `ConvertIn` and `ConvertOut` member functions to convert between UTF-8 and UCS-32.

6. The `UnicodeConverter` class can also throw exceptions if something goes wrong (for example, invalid UTF-8 input). See the source for details.