\blah{TeX}

Blahtex and Blahtexml version 0.9 manual

David Harvey and Gilles Van Assche

1 Introduction

This is the manual for blahtex and blahtexml version 0.9. The most up-to-date information about blahtex and blahtexml is available at

`http://gva.noekeon.org/blahtexml/`.

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 blahtexml command-line application** (Section 4) describes how to compile, install, and run the blahtexml command-line application.
- **The blahtex API** (Section 5) describes how to link blahtex directly into your code, which might give better performance in some environments.
- **History/changelog** (Section 6) summarises previous versions and changes.

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 (LATEX and `dvipng`).

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, including almost all of the symbols. A complete list of supported and quasi-supported commands can be found in Section 2.

Blahtex is internally Unicode-based. Non-ASCII characters may be used in text mode (e.g. within `\text{...}` blocks). These will be handled correctly for MathML output. For PNG output, blahtex can currently handle some extended Latin characters (see Section 2.22), and there is experimental support for Cyrillic and Japanese. More scripts may be added in the future.

Blahtex is open source software. The source code is released under the BSD 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 terms of the license.

Blahtex is written in C++. It compiles on Linux and Mac OS X systems, but probably is not as portable as it could be (see Section 3.1).

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 What is blahtexml?

Blahtexml is source-level superset of blahtex, so blahtexml does everything that blahtex does. In addition, blahtexml is also able to process a whole XML document into another XML document. Instead of converting only one formula at a time, blahtexml converts all the formulas of the given XML file into MathML.

Blahtexml requires one to have the XML parser library Xerces-C 2.x or 3.0 installed, to which blahtexml dynamically links. For those who do not need the blahtexml-specific functionality and/or do not have Xerces-C installed, the original blahtex can still be compiled.

Due to their big overlap, blahtex and blahtexml are maintained together.

1.4 The origin of the name 'blahtex'

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

1.5 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!)

1.6 Acknowledgements

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

Thanks to Jitse Niesen for his ongoing work on integrating blahtex into MediaWiki (currently on show at `wiki.blahtex.org`), 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 Macros

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 TeX only remembers the definition of `\abc` 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
```

2.3 Miscellaneous

`\sqrt` (including with optional argument)   `\substack`   `\overset`
`\underset`   `\not`

When it encounters `\not`, blahtex will attempt to find a MathML character that directly corresponds to the negation of any operator appearing after `\not`. Failing that, it will try to draw an ordinary slash in the right place, using the MathML `<mpadded>` element to fudge things.

2.4 Colour

Blahtex supports `\color{X}`, where `X` is one of the following named colours:

```
GreenYellow   Yellow   yellow   Goldenrod   Dandelion
Apricot   Peach   Melon   YellowOrange   Orange
BurntOrange  Bittersweet  RedOrange  Mahogany  Maroon
BrickRed   Red   red   OrangeRed   RubineRed
WildStrawberry  Salmon  CarnationPink  Magenta  magenta
VioletRed   Rhodamine   Mulberry   RedViolet   Fuchsia
Lavender   Thistle   Orchid   DarkOrchid   Purple   Plum
Violet  RoyalPurple  BlueViolet  Periwinkle  CadetBlue
```

```
CornflowerBlue  MidnightBlue  NavyBlue  RoyalBlue  Blue
blue  Cerulean  Cyan  cyan  ProcessBlue  SkyBlue
Turquoise  TealBlue  Aquamarine  BlueGreen  Emerald
JungleGreen  SeaGreen  Green  green  ForestGreen
PineGreen  LimeGreen  YellowGreen  SpringGreen
OliveGreen  RawSienna  Sepia  Brown  Tan  Gray  Black
black  White  white
```

At this time there is no support for colour models, so you can't do things like `\color[rgb]{0.2,0.3,0.4}`.

There are some subtle bugs in the parsing of `\color` commands. Things like `\overset{a}{\color{blue}x}` are not legal in LaTeX, for reasons I haven't yet fully investigated; blahtex still accepts them.

2.5 Text commands

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

The command `\hbox` doesn't really behave like it should, because MathML doesn't really have a notion of 'horizontal box'. Blahtex treats `\hbox` essentially equivalently to `\text`, with slightly different formatting rules. Things like `\hbox to 12pt` are not supported.

2.6 Fractions, binomials

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

2.7 Delimiters

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

2.8 Atom flavours

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

2.9 Limits

```
\limits  \nolimits  \displaylimits
```

2.10 Spacing

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

2.11 Accents

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

2.12 Fonts

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

2.13 Style

```
\displaystyle \textstyle \scriptstyle
\scriptscriptstyle
```

2.14 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
```

2.15 Escaped characters

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

2.16 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
```

2.17 Various mathematical symbols in no particular order

```
\ast \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 \circledR \yen
\maltese \circledS \Bbbk \jmath \ulcorner
\urcorner \llcorner \lrcorner \dashrightarrow
\dashleftarrow \backprime \vartriangle \blacktriangle
\triangledown \blacktriangledown \blacksquare
\lozenge \blacklozenge \bigstar \sphericalangle
\measuredangle \dotplus \ltimes \rtimes \Cap
\leftthreetimes \rightthreetimes \Cup \barwedge
\curlywedge \veebar \curlyvee \doublebarwedge
\boxminus \circleddash \boxtimes \circledast \boxdot
\circledcirc \boxplus \centerdot \divideontimes
\intercal \leqq \geqq \leqslant \geqslant
\eqslantless \eqslantgtr \gtrsim \lessapprox
\gtrapprox \approxeq \eqsim \lessdot \gtrdot \lll
\ggg \lessgtr \gtrless \lesseqgtr \gtreqless
\lesseqqgtr \gtreqqless \doteqdot \eqcirc
\risingdotseq \circeq \fallingdotseq \triangleq
\backsim \thicksim \backsimeq \thickapprox
```

```
\subseteqq \supseteqq \Subset \Supset \preccurlyeq
\succcurlyeq \curlyeqprec \curlyeqsucc \precsim
\succsim \precapprox \succapprox \Vvdash \shortmid
\shortparallel \bumpeq \between \Bumpeq \varpropto
\backepsilon \blacktriangleleft \blacktriangleright
\therefore \because \ngtr \nleqslant \ngeqslant
\nleqq \ngeqq \lneqq \gneqq \lvertneqq \gvertneqq
\lnsim \gnsim \lnapprox \gnapprox \nprec \nsucc
\npreceq \nsucceq \precneqq \succneqq \precnsim
\succnsim \precnapprox \succnapprox \nsim \ncong
\nshortmid \nshortparallel \nmid \nparallel \nvdash
\nvDash \nVdash \nVDash \ntriangleleft
\ntriangleright \ntrianglelefteq \ntrianglerighteq
\nsubseteq \nsupseteq \nsubseteqq \nsupseteqq
\subsetneq \supsetneq \varsubsetneq \varsupsetneq
\subsetneqq \supsetneqq \varsubsetneqq \varsupsetneqq
\leftleftarrows \rightrightarrows \leftrightarrows
\rightleftarrows \Lleftarrow \Rrightarrow
\twoheadleftarrow \twoheadrightarrow \leftarrowtail
\rightarrowtail \looparrowleft \looparrowright
\leftrightharpoons \rightleftharpoons \curvearrowleft
\curvearrowright \circlearrowleft \circlearrowright
\Lsh \Rsh \upuparrows \downdownarrows \multimap
\rightsquigarrow \leftrightsquigarrow \nLeftarrow
\nRightarrow \nleftrightarrow \nLeftrightarrow
\pitchfork \nexists \lhd \rhd \unlhd \unrhd
\leadsto \uplus \diamond \bigtriangleup
\bigtriangledown \ominus \oslash \odot \bigcirc
\amalg \prec \succ \preceq \succeq \dashv \asymp
\doteq \parallel \bowtie \surd \doublecap
\restriction \llless \gggtr \Doteq \doublecup
\dasharrow \vartriangleleft \vartriangleright \Join
```

2.18 Large operators

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

2.19 Symbols only available in text mode

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

2.20 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.

2.21 Unicode symbol translation in math mode

In math mode, blahtex accepts a number of non-ASCII symbols just like their command counterpart. These symbols are translated as TeX commands, as detailed in the table below. For instance, the character α (Unicode 0x3B1) is equivalent to the ASCII sequence `\alpha`. The benefit is input formulas that are more compact and more readable, provided that the file encoding and/or console character set allows for it. Note that this applies to both blahtex and blahtexml; see Section 4.3.6.

Symbol	Unicode	Translated as
$\neg$	000000AC	`\lnot`
$\pm$	000000B1	`\pm`
$\mathring{A}$	000000C5	`\AA`
$\times$	000000D7	`\times`
$\div$	000000F7	`\div`
Γ	00000393	`\Gamma`
Δ	00000394	`\Delta`
Θ	00000398	`\Theta`
Λ	0000039B	`\Lambda`
Ξ	0000039E	`\Xi`
Π	000003A0	`\Pi`
Σ	000003A3	`\Sigma`
Υ	000003A5	`\Upsilon`
Φ	000003A6	`\Phi`
Ψ	000003A8	`\Psi`
Ω	000003A9	`\Omega`
α	000003B1	`\alpha`
β	000003B2	`\beta`
γ	000003B3	`\gamma`
δ	000003B4	`\delta`
ε	000003B5	`\varepsilon`
ζ	000003B6	`\zeta`
η	000003B7	`\eta`
θ	000003B8	`\theta`
ι	000003B9	`\iota`
κ	000003BA	`\kappa`
λ	000003BB	`\lambda`
μ	000003BC	`\mu`

Symbol	Unicode	Translated as
ν	000003BD	\nu
ξ	000003BE	\xi
π	000003C0	\pi
ρ	000003C1	\rho
ς	000003C2	\varsigma
σ	000003C3	\sigma
τ	000003C4	\tau
υ	000003C5	\upsilon
φ	000003C6	\varphi
χ	000003C7	\chi
ψ	000003C8	\psi
ω	000003C9	\omega
ϑ	000003D1	\vartheta
ϕ	000003D5	\phi
ϖ	000003D6	\varpi
$\digamma$	000003DD	\digamma
$\varkappa$	000003F0	\varkappa
ϱ	000003F1	\varrho
ϵ	000003F5	\epsilon
$\backepsilon$	000003F6	\backepsilon
$\dagger$	00002020	\dagger
$\ddagger$	00002021	\ddagger
$\bullet$	00002022	\bullet
$\dots$	00002026	\dots
$\prime$	00002032	\prime
$\backprime$	00002035	\backprime
$\leftarrow$	00002190	\leftarrow
$\uparrow$	00002191	\uparrow
$\rightarrow$	00002192	\rightarrow
$\downarrow$	00002193	\downarrow
$\leftrightarrow$	00002194	\leftrightarrow
$\updownarrow$	00002195	\updownarrow
$\nwarrow$	00002196	\nwarrow
$\nearrow$	00002197	\nearrow
$\searrow$	00002198	\searrow
$\swarrow$	00002199	\swarrow
$\nleftarrow$	0000219A	\nleftarrow
$\nrightarrow$	0000219B	\nrightarrow
$\rightsquigarrow$	0000219D	\rightsquigarrow
$\twoheadleftarrow$	0000219E	\twoheadleftarrow
$\twoheadrightarrow$	000021A0	\twoheadrightarrow
$\leftarrowtail$	000021A2	\leftarrowtail
$\rightarrowtail$	000021A3	\rightarrowtail

Symbol	Unicode	Translated as
↦	000021A6	\mapsto
↩	000021A9	\hookleftarrow
↪	000021AA	\hookrightarrow
↫	000021AB	\looparrowleft
↬	000021AC	\looparrowright
↭	000021AD	\leftrightsquigarrow
↮	000021AE	\nleftrightarrow
↰	000021B0	\Lsh
↱	000021B1	\Rsh
↶	000021B6	\curvearrowleft
↷	000021B7	\curvearrowright
↺	000021BA	\circlearrowleft
↻	000021BB	\circlearrowright
↼	000021BC	\leftharpoonup
↽	000021BD	\leftharpoondown
↾	000021BE	\upharpoonright
↿	000021BF	\upharpoonleft
⇀	000021C0	\rightharpoonup
⇁	000021C1	\rightharpoondown
⇂	000021C2	\downharpoonright
⇃	000021C3	\downharpoonleft
⇄	000021C4	\rightleftarrows
⇆	000021C6	\leftrightarrows
⇇	000021C7	\leftleftarrows
⇈	000021C8	\upuparrows
⇉	000021C9	\rightrightarrows
⇊	000021CA	\downdownarrows
⇋	000021CB	\leftrightharpoons
⇌	000021CC	\rightleftharpoons
⇍	000021CD	\nLeftarrow
⇎	000021CE	\nLeftrightarrow
⇏	000021CF	\nRightarrow
⇐	000021D0	\Leftarrow
⇑	000021D1	\Uparrow
⇒	000021D2	\Rightarrow
⇓	000021D3	\Downarrow
⇔	000021D4	\Leftrightarrow
⇕	000021D5	\Updownarrow
⇚	000021DA	\Lleftarrow
⇛	000021DB	\Rrightarrow
⇝	000021DD	\leadsto
∀	00002200	\forall
∁	00002201	\complement

Symbol	Unicode	Translated as
∃	00002203	\exists
∄	00002204	\nexists
∇	00002207	\nabla
∈	00002208	\in
∉	00002209	\notin
∋	0000220B	\ni
∏	0000220F	\prod
∐	00002210	\coprod
∑	00002211	\sum
∓	00002213	\mp
∔	00002214	\dotplus
∘	00002218	\circ
√	0000221A	\surd
∝	0000221D	\propto
∠	00002220	\angle
∡	00002221	\measuredangle
∢	00002222	\sphericalangle
∤	00002224	\nmid
∥	00002225	\parallel
∦	00002226	\nparallel
∧	00002227	\wedge
∨	00002228	\vee
∩	00002229	\cap
∪	0000222A	\cup
∫	0000222B	\int
∬	0000222C	\iint
∭	0000222D	\iiint
∮	0000222E	\oint
∴	00002234	\therefore
∵	00002235	\because
∼	0000223C	\sim
∽	0000223D	\backsim
≀	00002240	\wr
≁	00002241	\nsim
≂	00002242	\eqsim
≃	00002243	\simeq
≅	00002245	\cong
≇	00002247	\ncong
≈	00002248	\approx
≊	0000224A	\approxeq
≎	0000224E	\Bumpeq
≏	0000224F	\bumpeq
≐	00002250	\doteq

Symbol	Unicode	Translated as
≑	00002251	\doteqdot
≒	00002252	\fallingdotseq
≓	00002253	\risingdotseq
≖	00002256	\eqcirc
≗	00002257	\circeq
≜	0000225C	\triangleq
≠	00002260	\neq
≡	00002261	\equiv
≤	00002264	\leq
≥	00002265	\geq
≦	00002266	\leqq
≧	00002267	\geqq
≨	00002268	\lneqq
≩	00002269	\gneqq
≪	0000226A	\ll
≫	0000226B	\gg
≬	0000226C	\between
≮	0000226E	\nless
≯	0000226F	\ngtr
≰	00002270	\nleq
≱	00002271	\ngeq
≲	00002272	\lesssim
≳	00002273	\gtrsim
≶	00002276	\lessgtr
≷	00002277	\gtrless
≺	0000227A	\prec
≻	0000227B	\succ
≼	0000227C	\preccurlyeq
≽	0000227D	\succcurlyeq
≾	0000227E	\precsim
≿	0000227F	\succsim
⊀	00002280	\nprec
⊁	00002281	\nsucc
⊂	00002282	\subset
⊃	00002283	\supset
⊆	00002286	\subseteq
⊇	00002287	\supseteq
⊈	00002288	\nsubseteq
⊉	00002289	\nsupseteq
⊊	0000228A	\subsetneq
⊋	0000228B	\supsetneq
⊎	0000228E	\uplus
⊏	0000228F	\sqsubset

Symbol	Unicode	Translated as
⊐	00002290	\sqsupset
⊑	00002291	\sqsubseteq
⊒	00002292	\sqsupseteq
⊓	00002293	\sqcap
⊔	00002294	\sqcup
⊕	00002295	\oplus
⊖	00002296	\ominus
⊗	00002297	\otimes
⊘	00002298	\oslash
⊙	00002299	\odot
⊚	0000229A	\circledcirc
⊛	0000229B	\circledast
⊝	0000229D	\circleddash
⊞	0000229E	\boxplus
⊟	0000229F	\boxminus
⊠	000022A0	\boxtimes
⊡	000022A1	\boxdot
⊢	000022A2	\vdash
⊣	000022A3	\dashv
⊤	000022A4	\top
⊥	000022A5	\bot
⊧	000022A7	\models
⊨	000022A8	\vDash
⊩	000022A9	\Vdash
⊪	000022AA	\Vvdash
⊬	000022AC	\nvdash
⊭	000022AD	\nvDash
⊮	000022AE	\nVdash
⊯	000022AF	\nVDash
⊲	000022B2	\lhd
⊳	000022B3	\rhd
⊴	000022B4	\unlhd
⊵	000022B5	\unrhd
⊸	000022B8	\multimap
⊺	000022BA	\intercal
⊻	000022BB	\veebar
⋀	000022C0	\bigwedge
⋁	000022C1	\bigvee
⋂	000022C2	\bigcap
⋃	000022C3	\bigcup
⋄	000022C4	\diamond
⋅	000022C5	\cdot
⋆	000022C6	\star

Symbol	Unicode	Translated as
⋇	000022C7	\divideontimes
⋈	000022C8	\bowtie
⋉	000022C9	\ltimes
⋊	000022CA	\rtimes
⋋	000022CB	\leftthreetimes
⋌	000022CC	\rightthreetimes
⋍	000022CD	\backsimeq
⋎	000022CE	\curlyvee
⋏	000022CF	\curlywedge
⋐	000022D0	\Subset
⋑	000022D1	\Supset
⋒	000022D2	\Cap
⋓	000022D3	\Cup
⋔	000022D4	\pitchfork
⋖	000022D6	\lessdot
⋗	000022D7	\gtrdot
⋘	000022D8	\lll
⋙	000022D9	\ggg
⋚	000022DA	\lesseqgtr
⋛	000022DB	\gtreqless
⋞	000022DE	\curlyeqprec
⋟	000022DF	\curlyeqsucc
⋦	000022E6	\lnsim
⋧	000022E7	\gnsim
⋨	000022E8	\precnsim
⋩	000022E9	\succnsim
⋪	000022EA	\ntriangleleft
⋫	000022EB	\ntriangleright
⋬	000022EC	\ntrianglelefteq
⋭	000022ED	\ntrianglerighteq
⋮	000022EE	\vdots
⋯	000022EF	\cdots
⋱	000022F1	\ddots
⌅	00002305	\barwedge
⌆	00002306	\doublebarwedge
⌈	00002308	\lceil
⌉	00002309	\rceil
⌊	0000230A	\lfloor
⌋	0000230B	\rfloor
⌜	0000231C	\ulcorner
⌝	0000231D	\urcorner
⌞	0000231E	\llcorner

Symbol	Unicode	Translated as
⌟	0000231F	\lrcorner
⌢	00002322	\frown
⌣	00002323	\smile
〈	00002329	\langle
〉	0000232A	\rangle
□	000025A1	\square
△	000025B3	\triangle
▲	000025B4	\blacktriangle
▵	000025B5	\vartriangle
▶	000025B6	\blacktriangleright
▹	000025B9	\triangleright
▽	000025BD	\bigtriangledown
▼	000025BE	\blacktriangledown
▿	000025BF	\triangledown
◀	000025C0	\blacktriangleleft
◃	000025C3	\triangleleft
◊	000025CA	\lozenge
◯	000025EF	\bigcirc
■	000025FC	\blacksquare
★	00002605	\bigstar
♠	00002660	\spadesuit
♣	00002663	\clubsuit
♡	00002665	\heartsuit
♢	00002666	\diamondsuit
♭	0000266D	\flat
♮	0000266E	\natural
♯	0000266F	\sharp
✓	00002713	\checkmark
⇠	0000290E	\dashleftarrow
⇢	0000290F	\dashrightarrow
⧫	000029EB	\blacklozenge
⨀	00002A00	\bigodot
⨁	00002A01	\bigoplus
⨂	00002A02	\bigotimes
⨄	00002A04	\biguplus
⨆	00002A06	\bigsqcup
⨌	00002A0C	\iiiint
⨿	00002A3F	\amalg
⩽	00002A7D	\leqslant
⩾	00002A7E	\geqslant
⪅	00002A85	\lessapprox
⪆	00002A86	\gtrapprox
⪉	00002A89	\lnapprox

Symbol	Unicode	Translated as
⪊	00002A8A	`\gnapprox`
⪋	00002A8B	`\lesseqqgtr`
⪌	00002A8C	`\gtreqqless`
⪕	00002A95	`\eqslantless`
⪖	00002A96	`\eqslantgtr`
⪯	00002AAF	`\preceq`
⪰	00002AB0	`\succeq`
⪵	00002AB5	`\precneqq`
⪶	00002AB6	`\succneqq`
⪷	00002AB7	`\precapprox`
⪸	00002AB8	`\succapprox`
⪹	00002AB9	`\precnapprox`
⪺	00002ABA	`\succnapprox`
⫅	00002AC5	`\subseteqq`
⫆	00002AC6	`\supseteqq`
⫋	00002ACB	`\subsetneqq`
⫌	00002ACC	`\supsetneqq`

2.22 Non-ASCII characters in text mode

Blahtex will serenely transcribe any non-ASCII characters for MathML output, as long as they appear in text mode (for example, surrounded by `\text{...}`). For PNG output, things are more difficult, because LaTeX needs special packages and fonts available. At a minimum, the blahtex command line option `--use-ucs-package` must be used. The following sections describe which characters are permitted for PNG output.

2.22.1 Extended Latin

The following characters are handled directly by the LaTeX `ucs` package.

¡ £ § © ¬ ® ° 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 Ǣ ǣ Ǧ ǧ Ǩ ǩ ǰ Ǵ ǵ Ǹ ǹ Ǽ ǽ
Ǿ ǿ Ș ș Ț ț Ȟ ȟ Ȧ ȧ Ȩ ȩ Ȯ ȯ Ȳ ȳ

Currently blahtex does not recognise TeX's accent commands (like `\"o`), so it is necessary to enter characters requiring accents directly in UTF-8.

2.22.2 Cyrillic

Blahtex experimentally supports Cyrillic characters, by using LaTeX's `fontenc` package with the `X2` font encoding. Input must be entered in UTF-8, and surrounded by the (nonstandard) `\cyr{...}` command. Commands like `\CYRSHA` are not supported. Only the basic Cyrillic alphabet is supported, which as far as I can tell is sufficient for Russian.

Disclaimer: I don't know anything about Cyrillic, or any languages that use it. If I've messed something up, your advice would be appreciated.

2.22.3 Japanese

Blahtex experimentally supports Japanese (Kanji, Hiragana, Katakana) by using the LaTeX `CJK` package. Input must be entered in UTF-8, and surrounded by the (nonstandard) `\jap{...}` command. The command-line option `--use-cjk-package` must be used. Additionally, the TeX system must have a Japanese font installed, and blahtex needs to be informed via the command-line option `--japanese-font`.

Disclaimer: I don't know anything about the Japanese language or writing system. If I've messed something up, your advice would be appreciated.

2.23 Partial list of differences between blahtex and texvc

2.23.1 Additional commands

Blahtex supports many TeX/LaTeX/AMS-LaTeX commands not supported by texvc, especially many of the symbols in AMS-LaTeX.

2.23.2 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.

2.23.3 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.

2.23.4 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.

2.23.5 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 (assuming that the `--texvc-compatible-commands` option is active).

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 a public keyserver.

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

3.1 System prerequisites

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

- Linux with gcc 4.0.2 20050808 (prerelease) or with gcc 4.3.1
- Mac OS 10.4.5 (PowerPC) with gcc 4.0.1

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. It should be fine with 512MB or above.

Other UNIX-based systems might work too. You will probably encounter problems with compilers other than gcc, or with older versions of gcc. (Probably gcc 3.3 is still okay.) I have personally met at least one older Solaris compiler that couldn't stomach the code. Your compiler must support `wstring` and 32-bit `wchar_t`s. If you want to compile it on MS Windows... good luck, let me know how it goes.

You will need an installation of the GNU `iconv` library. 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).

3.1.1 Prerequisites for generating PNG output

To generate PNGs, you will need LaTeX and the `dvipng` utility, which is included in many LaTeX distributions. Blahtex assumes that the following LaTeX packages are available: `color`, `fontenc`, `inputenc`, `amsmath`, `amsfonts`, `amssymb`. All of these packages are included in teTeX, one of the most popular TeX distributions for UNIX systems.

Additionally, to handle non-ASCII characters, the `ucs` package must be installed, and blahtex must be informed by using the `--use-ucs-package` command line option. To enable computation of height and depth of the output PNG image, the `preview` package must be installed, and blahtex must be informed by using the `--use-preview-package` option.

3.1.2 Modified version of `dvipng`

The version of `dvipng` running on the blahtex website is a slightly modified version of `dvipng` 1.7. The modification pertains to the automatic hinting method used with the underlying FreeType 2 library, and was made with the help of the author of `dvipng`, Jan-Åke Larsson (thanks Jan-Åke!).

It's quite simple: in the source file `ft.c`, just replace `FT_LOAD_NO_HINTING` by `FT_LOAD_TARGET_LIGHT`, and recompile. The author has indicated that this modification will appear in `dvipng` version 1.8.

3.1.3 Prerequisites for Japanese in PNG output

To handle Japanese, the LaTeX `CJK` package must be installed, and a Japanese font must be installed.

Warning: Installing TrueType CJK fonts for use by LaTeX/dvipng is a dark art. In this section I will describe a sequence of steps that worked for me. I will explain along the way what I believe the purpose of each step to be, and caveats *that you should be aware of.* ***However, this should not be construed to imply that I have any idea at all of what I am talking about.***

You will need a Japanese TrueType font. For testing, I have been using the Sazanami gothic font: `http://sourceforge.jp/projects/efont/files/`. Look inside for the TrueType font file `sazanami-gothic.ttf`.

Warning: I have not read the license document for this font. It is mostly in Japanese. It is quite possible that it is ***not legal*** *to use this font for certain purposes. Since it is advertised as being targeted at OpenOffice, I expect that all is okay, but* ***I am not a lawyer.***

The strategy outlined below is to convert the TrueType font to a bunch of smaller Type 1 fonts, and to provide enough other information to make LaTeX and `dvipng` happy.

You will need FontForge, from `http://fontforge.sourceforge.net/`. (Note that to install FontForge on Mac OS X, you will need the StuffIt Expander utility to decompress the installation package. StuffIt Expander was included in Mac OS 10.3.x, but is not shipped with Mac OS 10.4.x. I had a copy available from an older OS, but if you have only OS 10.4.x, you will need to download StuffIt Expander from `http://www.stuffit.com/mac/expander/`. Also on the Mac you need to make sure that you have an X11 server available. On Mac OS 10.4.x it should be pre-installed in `/Applications/Utilities/X11.App`. On earlier versions you may need to download X11 from Apple's website.)

Create a temporary working directory somewhere, which I will refer to in these instructions as `/temp`.

You need to select a name for your font. Probably best to keep it very short. I will use the name 'saza' throughout the following example; you will need to replace every 'saza' with whatever you have chosen.

Boot up X11, and run FontForge. You should get an 'Open Font' dialog; open the `ttf` file from above. Then select 'Generate Fonts...' from the File menu. Navigate to your `/temp` directory; this is where the output from the 'generate fonts' process will be saved. On the drop-down list on the left, select 'PS Type 1 (Multiple)'. (The point here is to split the font up into many smaller sub-fonts. This is necessary because TeX can only really work with fonts that contain at most 256 symbols, and CJK fonts have many more than that.) The default file name will be something like `sazanami-gothic%s.pfb`; change this to `saza-uni%s.pfb`. Now press 'Options', and make sure 'Output TFM & ENC' is checked. Then hit 'Save'. A new 'Find Sub Font Definitions' dialog will pop up. You will need to find the file `Unicode.sfd` on the web somewhere (Google is your friend); save this file somewhere and tell the dialog where it is. Press OK.

FontForge should go away and think for a while. When it's finished, your `/temp` directory should be filled with lots of `.tfm`, `.pfb`, `.afm`, and `.enc` files. You can throw away the last two; we only need the `.tfm` and `.pfb` files. In your `texmf` tree, make a new directory called `/texmf/fonts/tfm/saza/`, and put all the `.tfm` files there. Similarly, put all the `.pfb` files into a directory `/texmf/fonts/type1/saza/`.

(The `.tfm` files are 'TeX font metric' files. Roughly speaking, they tell TeX how much space each character takes up. The corresponding `.pfb` files are Adobe Type 1 font files; they describe the actual glyphs for each character.)

Create a plain text file called `C70saza.fd`, and fill it with the following text:

```
\DeclareFontFamily{C70}{saza}{\hyphenchar \font\mne}
\DeclareFontShape{C70}{saza}{m}{n}{<-> CJK * saza-uni}{}
\DeclareFontShape{C70}{saza}{bx}{n}{<-> CJKb * saza-uni}{CJKbold}
```

Save this file under `/texmf/tex/latex/saza/`. (I think the idea of this file is to tell LaTeX something about the new font you have installed.)

That's all the files you need. Now you need to run `mktexlsr` (or `sudo mktexlsr`) to update TeX's filename cache.

When you run blahtex, you will need to use the command line options `--use-cjk-package --use-ucs-package --japanese-font saza`.

3.2 Compiling blahtex

Unpack the source into your favourite directory.

- 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`.

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 2.23.5).
- `--print-error-messages`. This will print out a list of all error IDs and corresponding messages that blahtex can possibly emit inside an `<error>` block (see Section 3.4).
- `--displaymath`. This tells blahtex to render the formula in "display math," for full-size MathML or PNGs displayed on their own line. Without this option, the formula is rendered in "inline math".

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` (default). 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`. 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 2.20).

Blahtex pays a lot of attention to spacing, because the MathML defaults (via the operator dictionary) are often inadequate. 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.
- `--use-ucs-package`. This tells blahtex it may use the LaTeX `ucs` package to handle non-ASCII characters. Obviously, it is necessary to install the `ucs` package before using this option. See Section 2.22 for more information.
- `--use-cjk-package`. This tells blahtex it may use the LaTeX `CJK` package to handle Chinese/Japanese/Korean characters. Obviously, it is necessary to install the `CJK` package before using this option. See also Section 3.1.3.
- `--use-preview-package`. This tells blahtex it may use the LaTeX `preview` package. Obviously, it is necessary to install the `preview` package before using this option. With this option enabled, blahtex is able to compute the height and depth of the output PNG image (via dvipng).
- `--japanese-font` *`fontname`*. Specifies which font to use for characters surrounded by `\jap{...}`. See also Section 3.1.3.
- `--shell-latex` *`command`*. Specifies the command to use for running LaTeX. Default is just `latex`.

- `--shell-dvipng` *`command`*. Specifies the command to use for running dvipng. Default is just `dvipng`.
- `--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.
- `--png-latex-preamble` *`content`*. Specifies LaTeX content that is inserted before `\begin{document}` in the generated `.tex` file. This can be used, for instance, to include an additional package, e.g., `\usepackage{mathpazo}`.
- `--png-latex-before-math` *`content`*. Specifies LaTeX content that is inserted just before the equation (after `\begin{document}`) in the generated `.tex` file. This can be used, for instance, to select a particular font, e.g., `\fontsize{20}{0}\selectfont`.

3.3.4 Debugging options

- `--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.
- `--keep-temp-files`. Instructs blahtex not to delete any of the temporary files that get created during PNG generation.

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:

  - `InvalidUtf8Input`
  - `IllegalCharacter`
  - `TooManyTokens`
  - `NonAsciiInMathMode`
  - `ReservedCommand`
  - `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`
  - `MissingDelimiter`
  - `IllegalDelimiter`
  - `MisplacedLimits`
  - `DoubleSuperscript`
  - `DoubleSubscript`
  - `AmbiguousInfix`
  - `InvalidColour`

- 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 IDs that can occur here are:
    - `TooManyMathmlNodes`
    - `UnavailableSymbolFontCombination`
  - 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 the option `--use-preview-package` was used, the `<png>` block will also contain blocks `<height>H</height>` and `<depth>D</depth>` which indicate the height and depth of the image, in pixels. (These are computed by `dvipng`.) If you want to display the PNG in a web page so that it is aligned with surrounding text, you can use the depth value as follows: `<img src="..." style="vertical-align: -`*D*`px">`.

    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:
    - `CannotCreateTexFile`
    - `CannotWriteTexFile`
    - `CannotRunLatex`

- ∗ CannotRunDvipng
- ∗ CannotWritePngDirectory
- ∗ CannotChangeDirectory
- ∗ LatexPackageUnavailable
- ∗ WrongFontEncoding
- ∗ WrongFontEncodingWithHint
- ∗ IllegalNestedFontEncodings
- ∗ LatexFontNotSpecified
- ∗ PngIncompatibleCharacter

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 blahtexml command-line application

The blahtexml source code is available from `http://gva.noekeon.org/blahtexml`.

4.1 System prerequisites

In addition to the prerequisites of blahtex (see Section 3.1), blahtexml requires one to have Xerces-C 2.x or 3.0 installed. Xerces-C is an XML parser library and is available at `http://xerces.apache.org/xerces-c/`. Blahtexml dynamically links to Xerces-C.

4.2 Compiling blahtexml

Unpack the source into your favourite directory.

- If you're running Linux, just type `make blahtexml-linux`.
- If you're running Mac OS X, try `make blahtexml-mac`.

You should then find an executable `blahtexml` in the current directory.

4.3 Using blahtexml

Blahtexml contains blahtex, which means that all the command-line options of blahtex are available with blahtexml. They are described in Section 3.3.

What is specific to blahtexml is the `--xmlin` option. This tells blahtexml to input an XML file and to convert all the equations it finds into an output XML file, which contains the equivalent MathML code. All the elements, attributes and processing instructions are copied from the input to the output XML file, unchanged. When it encounters an equation in blahtex, it is converted into MathML.

When used, the `--xmlin` option must be first. Note that, in this case, not all the blahtex command line options work. The options that are ignored when `--xmlin` is used are: `--png`, `--mathml-encoding`, `--other-encoding` and `--disallow-plane-1`.

In the following, we describe how blahtexml locates blahtex formulas and how the process works exactly. For this, we assume that the reader has some familiarity with the XML syntax and with the XML namespaces.

In an XML file, blahtexml looks for attributes with name `m`, `inline` or `block` in the namespace `http://gva.noekeon.org/blahtexml`. It will then remove this attribute and expand the produced MathML inside the element that contains the attribute. Let us just illustrate this with an example.

Consider the following input file:

```
<?xml version="1.0"?>
<equations xmlns:b="http://gva.noekeon.org/blahtexml">
  <equation b:inline="x+y"/>
  <equation b:block="\exp(-\gamma x)"/>
</equations>
```

By calling `blahtexml --xmlin < example1.xml`, blahtexml will produce the following output, where for clarity some MathML elements are not written:

```
<?xml version="1.0" encoding="UTF-8"?>
<equations xmlns:b="http://gva.noekeon.org/blahtexml">
  <equation>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <mi>x</mi>
      <mo lspace="0.222em" rspace="0.222em">+</mo>
      <mi>y</mi>
    </math>
  </equation>
  <equation>
    <math xmlns="http://www.w3.org/1998/Math/MathML"
        display="block">
      <mstyle displaystyle="true">
      <mi>exp</mi>[...]
    </math>
  </equation>
</equations>
```

As one can see in this example, the `inline` attribute produces MathML in inline mode (the default of MathML), while the `block` attribute produces MathML in block mode by adding the attribute `display="block"` in the `math` element. When in block mode, the display style is set; this causes, e.g., the sum boundaries to be moved under and over the sum sign.

The `m` element does not create a `math` element, but instead puts the MathML content as is. This can be useful if, e.g., one wants to type an equation partly in MathML and partly in blahtex. This is illustrated in the next example, where a blahtex equation is given inside a `msqrt` MathML element. The input file

```
<root xmlns:b="http://gva.noekeon.org/blahtexml">
  <math xmlns="http://www.w3.org/1998/Math/MathML">
    <msqrt b:m="x+y"/>
  </math>
</root>
```

yields as output:

```
<root xmlns:b="http://gva.noekeon.org/blahtexml">
  <math xmlns="http://www.w3.org/1998/Math/MathML">
    <msqrt>
      <mi>x</mi>
      <mo lspace="0.222em" rspace="0.222em">+</mo>
      <mi>y</mi>
    </msqrt>
  </math>
</root>
```

Note that if more than one attribute in the blahtex namespace are present, only one is processed, with `m` having the highest priority, then `inline` and finally `block`.

4.3.1 Annotating with TEX format

In parallel to the formula expressed in MathML, the standard allows annotations in other formats. By using the command line option `--annotate-TeX`, blahtexml produces the formula in TEX/LATEX format in an `annotation` tag, in addition to the MathML format. This way, the output can also be used as the basis to produce a TEX or LATEX file from the same source in XML.

For instance, consider the following input file:

```
<?xml version="1.0"?>
<equations xmlns:b="http://gva.noekeon.org/blahtexml">
  <equation b:inline="x+y"/>
</equations>
```

With --annotate-TeX, the output will be:

```
<?xml version="1.0" encoding="UTF-8"?>
<equations xmlns:b="http://gva.noekeon.org/blahtexml">
  <equation>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <semantics>
        <mrow>
          <mi>x</mi>
          <mo lspace="0.222em" rspace="0.222em">+</mo>
          <mi>y</mi>
        </mrow>
        <annotation encoding="TeX">x + y</annotation>
      </semantics>
    </math>
  </equation>
</equations>
```

4.3.2 Annotating with PNG images

Similarly, the command line option --annotate-PNG instructs blahtexml to produce PNG files in addition ot the MathML output. The file name is put in an annotation tag.

For instance, consider the following input file:

```
<?xml version="1.0"?>
<equations xmlns:b="http://gva.noekeon.org/blahtexml">
  <equation b:inline="x+y"/>
</equations>
```

With --annotate-TeX, the output will be:

```
<?xml version="1.0" encoding="UTF-8"?>
<equations xmlns:b="http://gva.noekeon.org/blahtexml">
  <equation>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <semantics>
        <mrow>
          <mi>x</mi>
          <mo lspace="0.222em" rspace="0.222em">+</mo>
          <mi>y</mi>
        </mrow>
        <annotation encoding="image-file-PNG">
            ./f05c46190061a618fd432bf5471cc2ab.png
        </annotation>
      </semantics>
    </math>
  </equation>
</equations>
```

Of course, the command line options `--annotate-TeX` and `--annotate-PNG` can be combined, producing two `annotation` tags in the output.

4.3.3 MathML namespace in output file

The MathML element produced in the output are in the MathML namespace, namely `http://www.w3.org/1998/Math/MathML`. There are two ways to express the namespace, either by adding the `xmlns` attribute to the outer MathML element, or by adding a prefix associated to the MathML namespace to all the MathML elements. By default, or using the `--mathml-nsprefix-auto` option, blahtexml automatically chooses between the two alternatives. Either a prefix already exists and blahtex reuses it, or such a prefix does not exist and an `xmlns` attribute is added.

From the point of view of XML namespaces, both approaches are equivalent. Nevertheless, some XML applications predate the introduction of XML namespaces and it may sometimes be necessary to force either solution.

- `--mathml-nsprefix-auto`. This is the default option: blahtexml automatically chooses to add a prefix or not.
- `--mathml-nsprefix-none`. The produced MathML elements are not prefixed. The `xmlns` attribute is added to the outer MathML element.
- `--mathml-nsprefix`. This option requires a parameter: the prefix (string). The produced MathML elements are prefixed with the given prefix and a colon.

Consider the following input file:

```
<root xmlns:b="http://gva.noekeon.org/blahtexml">
  <eq b:inline="x"/>
  <eq xmlns:m="http://www.w3.org/1998/Math/MathML" b:inline="x"/>
</root>
```

Invoking blahtexml using the default option `--mathml-nsprefix-auto`, one gets the following result:

```
<root xmlns:b="http://gva.noekeon.org/blahtexml">
  <eq>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <mi>x</mi>
    </math>
  </eq>
  <eq xmlns:m="http://www.w3.org/1998/Math/MathML">
    <m:math><m:mi>x</m:mi></m:math>
  </eq>
</root>
```

Using `--mathml-nsprefix-none`, one gets the following result:

```
<root xmlns:b="http://gva.noekeon.org/blahtexml">
  <eq>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <mi>x</mi>
    </math>
  </eq>
  <eq xmlns:m="http://www.w3.org/1998/Math/MathML">
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <mi>x</mi>
    </math>
  </eq>
</root>
```

And using `--mathml-nsprefix m`, one gets the following result:

```
<root xmlns:b="http://gva.noekeon.org/blahtexml">
  <eq>
    <m:math xmlns:m="http://www.w3.org/1998/Math/MathML">
      <m:mi>x</m:mi>
    </m:math>
  </eq>
  <eq xmlns:m="http://www.w3.org/1998/Math/MathML">
    <m:math><m:mi>x</m:mi></m:math>
  </eq>
</root>
```

4.3.4 Output document type

By default, the generated XML file does not contain a document type declaration. If the output file is intended to a given XML application, a `DOCTYPE` declaration may be needed. The `--doctype-*` command-line options provide a way to specify this.

- `--doctype-system`. This option takes a reference to a DTD (string) as argument and causes blahtexml to output a `SYSTEM` document type declaration with the given reference.

- `--doctype-public`. This option takes two arguments: a public ID (string) and a reference to a DTD (string). Blahtex produces a `PUBLIC` document type declaration with the given public ID and reference.

- `--doctype-xhtml+mathml`. This option is equivalent to `--mathml-nsprefix-none --doctype-public "-//W3C//DTD XHTML 1.1 plus MathML 2.0//EN" "http://www.w3.org/TR/MathML2/dtd/xhtml-math11-f.dtd"` and is useful to produce valid XHTML+MathML output.

4.3.5 Error reporting

If a blahtex equation given in the input XML file generates an error during its conversion to MathML, blahtexml adds an `error` element (in the blahtex namespace) instead of the MathML elements. The blahtex formula is not discarded, so that the user can more easily see what caused the problem. Furthermore, the number of errors encountered is reported on the screen.

For instance, the following input file

```
<root xmlns:b="http://gva.noekeon.org/blahtexml">
  <eq b:inline="\qwerty"/>
</root>
```

generates the following output file

```
<root xmlns:b="http://gva.noekeon.org/blahtexml">
  <eq b:inline="\qwerty">
    <error xmlns="http://gva.noekeon.org/blahtexml">
      Unrecognised command "\qwerty"
    </error>
  </eq>
</root>
```

4.3.6 Unicode symbol translation in math mode

As detailed in Section 2.21, blahtexml accepts some Unicode symbols and translates them into TEXcommands. For instance, the following three lines are equivalent and will give the same output:

- `<eq b:inline="\Phi \leq \Omega \approx \Gamma"/>`

- `<eq b:inline="`$\Phi \leq \Omega \approx \Gamma$`"/>`

- `<eq b:inline="&#x3A6;&#x2264;&#x3A9;&#x2248;&#x393;"/>`

The first line uses the traditional TEXcommands. The second line uses the Unicode symbols directly, assuming that the encoding of the XML file allows for it. Note that UTF-8, the default encoding in XML, includes all Unicode characters. The third line shows that it is also possible to use XML entities to input Unicode characters.

5 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.)

5.1 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.

5.2 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).

5.3 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.

6 History/changelog

- Version 0.1 (Jul/2005). You don't want to know about this one.
- Version 0.2 (2/Aug/2005). Initial public release.
- Version 0.2.1 (8/Aug/2005). Now compiles under Linux.
- Version 0.3.x (Aug 2005 to Jan 2006). Series of internal development releases, everything getting completely rewritten. It would be an act of irresponsibility to list every change.
- Version 0.4 (29/Jan/2006). Accompanies announcement of test wiki.
- Version 0.4.1 (8/Feb/2006). Added `--compute-vertical-shift` option.
- Version 0.4.2 (12/Feb/2006).
  - Greatly improved coverage of symbols in LaTeX and AMS-LaTeX.
  - Greatly improved coverage of `\not`.
  - Now `UnavailableSymbolFontCombination` and `InvalidNegation` errors are only flagged during MathML output; i.e. these errors no longer prevent PNG output.
  - Added `--keep-temp-files` option.
  - Fixed a PNG clipping bug in certain cases where dvips gets the PS bounding box incorrect. For example, when translating `\displaystyle \int`, half of the integral sign would go missing. (This bug affects texvc too.)
  - Changed behaviour of `<vshift>` block; now such a block appears even if the shift should be zero.
  - Fixed a few incorrect MathML characters.
- Version 0.4.3 (25/Feb/2006).
  - Now supports `\color`; added corresponding error code `InvalidColour`.
  - Numerous internal structural changes, especially an overhaul of the MathML output code.
  - Improved node merging heuristics, for things like `123^5`.
  - Corrected parsing of `\not`. Now blahtex will make a reasonable attempt on any `\not` that comes its way; the `InvalidNegation` error message has consequently been removed.
  - Fixed a bug that caused incorrect font attributes for input like `\rm \boldsymbol x`.
  - Added the `\ast` command (how did I ever miss that?)
- Version 0.4.4 (25/Mar/2006).

  - Changed default spacing mode from `moderate` to `strict`.
  - Changed from using dvips/ImageMagick to dvipng. Consequently the `--shell-dvips`, `--shell-convert` and `--convert-options` options have been removed, and replaced by `--shell-dvipng`. The error messages `CannotRunConvert` and `CannotRunDvips` have been removed and replaced by `CannotRunDvipng` and `CannotWritePngDirectory`.
  - Added flag `--use-preview-package`.
  - Removed the `--compute-vertical-shift` option; now the vertical shift is always computed (by dvipng) as long as the LaTeX `preview` package is loaded, but its name has been changed to 'depth'. Accordingly, the `<vshift>` output block has been replaced by `<height>` and `<depth>` blocks. The numbers themselves are now computed by dvipng, which is much neater and more reliable.
  - Added support for Cyrillic and Japanese in PNG output:
    - Added `--use-cjk-package` and `--japanese-font` options.
    - Added commands `\cyr` and `\jap`.
    - Added error messages:
      - `WrongFontEncoding`
      - `WrongFontEncodingWithHint`
      - `IllegalNestedFontEncodigs`
      - `LatexPackageUnavailable`
      - `LatexFontNotSpecified`
  - Corrected MathML characters for `\longrightarrow` and friends; however they are currently disabled because of poor font support.
  - Fixed spacing for `\substack` and the `aligned` environment. Note however that Firefox still doesn't support the requisite `rowspacing` and `columnspacing` attributes, so it won't look right yet in Firefox.
  - Changed format of `--print-error-messages` slightly.
  - Finished adding MathML character names for all commands added in version 0.4.2.

- Version blahtexml 0.4.4 (2/Nov/2007) by GVA
  - Added the blahtexml extension.

- Version blahtexml 0.5 (16/May/2008) by GVA
  - Added input symbol translation.
  - Improved makefile based on user feedback (Mac compilation, lower optimization level, documentation generation).

- Version blahtexml 0.6 (27/July/2008) by GVA
  - Fixed compilation issue with GCC 4.3.1.

  - PNGs can be rendered in ”display math” mode (patch by Ari Stern).
  - Fixed incorrect merging of identifiers such as `\hbar\Phi`.

- Version blahtexml 0.7 (3/Aug/2009)
  - License changes in agreement with both contributors:
    - The source code is released under the BSD license
    - The text of this manual is released under the Creative Commons Attribution license
  - For PNG output, LaTeX code can be inserted before purified TeX equations (e.g., to change fonts) (thanks to Mikkel Ricky)
  - Support for the Ångström symbol $\AA$ (`\AA`) (thanks to Paul Dlug)

- Version blahtexml 0.8 (8/Mar/2010)
  - Compatibility with Xerces-C 3.0 has been added.
  - Annotations with TeX and PNG outputs can be specified with command line options `--annotate-TeX` and `--annotate-PNG`.

- Version blahtexml 0.9 (27/Nov/2010)
  - The `--displaymath` option is now a general option, with an effect on MathML output (not only PNG).
  - With `--xmlin`, block equations (i.e, `<math display="block">`) are set in display style.