

L^AT_EX News, Issues 1–24

Contents

Issue 1, June 1994	2	Issue 6, December 1996	7
Welcome to L ^A T _E X News	2	Welcome to L ^A T _E X News 6	7
L ^A T _E X 2 _ε —the new L ^A T _E X release	2	Mono-case file names	7
Why a new L ^A T _E X?	2	Another input encoding	7
Processing documents with L ^A T _E X 2 _ε	2	Better user-defined math display environments	7
New packages	2	Docstrip improvements	7
Further information	2	AMS L ^A T _E X update	7
		Graphics package update	7
		EC Fonts released	7
Issue 2, December 1994	3	Issue 7, June 1997	8
Welcome to L ^A T _E X News 2	3	T1 encoded Computer Modern fonts	8
December 1994 release of L ^A T _E X	3	T1 encoded Concrete fonts	8
Accented input	3	Further input encodings	8
AMS-L ^A T _E X	3	Normalising spacing after punctuation	8
L ^A T _E X on the internet	3	Accessing Bold Math Symbols	8
Further information	3	Policy on standard classes	8
		New addresses for TUG	8
Issue 3, June 1995	4	Issue 8, December 1997	9
Welcome to L ^A T _E X News 3	4	New supported font encodings	9
June 1995 release of L ^A T _E X	4	New input encodings	9
Additional input encodings	4	Tools	9
L ^A T _E X getting smaller	4	Graphics	9
Distribution and modification	4	L ^A T _E X3 experimental programming conventions	9
AMS-L ^A T _E X full release	4	Issue 9, June 1998	10
PostScript fonts	4	New math font encodings	10
Further information	4	A new math accent	10
		Extended <code>\DeclareMathDelimiter</code>	10
Issue 4, December 1995	5	Tools distribution	10
Welcome to L ^A T _E X News 4	5	Support for Cyrillic encodings	10
L ^A T _E X getting smaller	5	Default docstrip header	10
New ‘concurrent’ docstrip	5	Issue 10, December 1998	11
New T1 encoded fonts	5	Five years of L ^A T _E X 2 _ε	11
More robust commands	5	Restructuring the L ^A T _E X distribution	11
New Interface to building ‘extension’ classes	5	L ^A T _E X Project on the Internet	11
More Input Encodings	5	Restructuring the L ^A T _E X package licenses	11
Further information	5	Support for Cyrillic encodings	11
		Tools distribution	11
Issue 5, June 1996	6	Issue 11, June 1999	12
Welcome to L ^A T _E X News 5	6	Back in sync	12
Extra possibilities for section headings	6	Yearly release cycles	12
The ‘openany’ option in the ‘book’ class	6	LPPL update	12
More font (output) encodings	6	The future of SliT _E X	12
More input encodings supported	6	Fontenc package peculiarities	12
Fixes and improvements	6	New math font encodings	12
Changes to the ‘tools’ packages	6	Tools distribution	12
New copy of the L ^A T _E X bug database	6	Coming soon	12

Issue 12, December 1999	13	Issue 19, September 2009	21
LPPL update	13	New L ^A T _E X release	21
fixltx2e package	13	New code repository	21
Outcome of TUG '99 (Vancouver)	13	Babel	21
		The future	21
Issue 13, June 2000	14	Issue 20, June 2011	22
Yearly release cycle	14	Scheduled L ^A T _E X bug-fix release	22
PSNFSS: Quote of the Month	14	Continued development	22
New AMS-L ^A T _E X	14	Release notes	22
New input encoding latin4	14	Font subsets covered by Latin Modern and	
New experimental code	14	TeX Gyre	22
		Redefinition of \enddocument	22
Issue 14, June 2001	15	Small improvement with split footnotes in	
Future releases	15	ftnright	22
New release of Babel (required)	15	Improvement in xspace and font-switching	22
New input encoding latin9	15	RTL in multicol	22
New tools	15	Improve French babel interaction with	
New experimental code	15	varioref	23
Issue 15, December 2003	16	Issue 21, May 2014	24
Anniversary release	16	Scheduled L ^A T _E X bug-fix release	24
LPPL – new version	16	Release notes	24
Small updates to varioref	16	fixltx2e updates	24
New and more robust commands	16	New fltrace package	24
Fixing font sizes	16	inputenc package updates	24
Font encodings	16	The tools directory	24
Displaying font tables	16	multicol updates	25
New input encodings	16	tabularx updates	25
Unicode input	16	showkeys updates	25
And finally ... pict2e	16	color updates	25
		graphicx updates	25
Issue 16, December 2003	17	keyval updates	25
Anniversary news	17	Standard L ^A T _E X (L ^A T _E X 2 _ε) and expl3	25
TLC2: The L ^A T _E X Companion – 2nd edition!	17	Issue 22, January 2015	26
Future maintenance	17	New L ^A T _E X 2 _ε bug-fix policy	26
LPPL certification	17	Introduction	26
Use of ε-TeX/pdfTeX	17	The latexrelease package	26
End of ‘autoload’ support	17	The \IncludeInRelease command	26
New models, new code	17	Limitations of the approach	26
Issue 17, December 2005	18	Updates to the kernel	27
Project licence news	18	Updates incorporated from fixltx2e	27
New guide on font encodings	18	ε-TeX register allocation	27
Robust commands in math	18	Additional L ^A T _E X float storage	27
Updates of required packages	18	Built-in support for Unicode engines	27
Work on L ^A T _E X fixes	18	l3build	27
The graphics bundle	19	Hyperlinked documentation and TDS zip files	27
Future development	19	Issue 23, October 2015	28
Issue 18, December 2007	20	Enhanced support for LuaTeX	28
		Names of LuaTeX primitive commands	28
		TeX commands for allocation in LuaTeX	28
		Predefined Lua functions	29
		Support for older releases and plain TeX	29

Additional LuaTeX support packages . . .	29
More Floats and Inserts	29
Updated Unicode data	29
Support for Comma Accent	29
Extended inputenc	29
Pre-release Releases	29
Updates in tools	29
Issue 24, February 2016	30
LuaTeX support	30
Unicode data	30
More support for east European accents	31
Changes in Graphics	31
Changes in Tools	31
Improving support for Unicode engines	31

L^AT_EX News

Issue 1, June 1994

Welcome to L^AT_EX News

An issue of *L^AT_EX News* will accompany every future release of L^AT_EX. It will tell you about important events, such as major bug fixes, newly available packages, or any other L^AT_EX news.

L^AT_EX 2_ε—the new L^AT_EX release

The most important news is the release of L^AT_EX 2_ε, the new version of the L^AT_EX software. This version has better support for fonts, graphics and colour, and will be actively maintained by the L^AT_EX3 project team. Upgrades will be issued every six months, in June and December.

Why a new L^AT_EX?

Over the years many extensions have been developed for L^AT_EX. This is, of course, a sure sign of its continuing popularity but it has had one unfortunate result: incompatible L^AT_EX formats came into use at different sites. Thus, to process documents from various places, a site maintainer was forced to keep L^AT_EX (with and without NFSS), S^LI^TE_X, $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX, and so on. In addition, when looking at a source file it was not always clear for which format the document was written.

To put an end to this unsatisfactory situation a new release of L^AT_EX was produced. It brings all such extensions back under a single format and thus prevents the proliferation of mutually incompatible dialects of L^AT_EX 2.09. The new release was available for several months as a test version, and the final release of 1 June officially replaces the old version.

Processing documents with L^AT_EX 2_ε

Documents written for L^AT_EX 2.09 will still be read by L^AT_EX 2_ε. Any such document is run in *L^AT_EX 2.09 compatibility mode*.

Unfortunately, compatibility mode comes with a price: it can run up to 50% slower than L^AT_EX 2.09 did. If you want to run your document in the faster *native mode*, you should try replacing the line:

```
\documentstyle[options,packages]{class}
```

with:

```
\documentclass[options]{class}
\usepackage{latexsym,packages}
```

Unfortunately, this will not always work, because some L^AT_EX 2.09 packages will only work in L^AT_EX 2_ε compatibility mode. You should find out if there is a L^AT_EX 2_ε version of the package available.

L^AT_EX 2_ε native mode also gives access to the new features of L^AT_EX 2_ε, described in *L^AT_EX 2_ε for authors*.

New packages

L^AT_EX 2_ε has much better support for graphics, colour, fonts, and multi-lingual typesetting. The following software should be available from the distributor who brought you L^AT_EX 2_ε:

babel, for typesetting in many languages.

color, for colour support.

graphics, for including images.

mfnfss, for using bitmap fonts.

psnfss, for using Type 1 fonts.

tools, other packages by the L^AT_EX3 team.

The packages come with full documentation, and are also described in *L^AT_EX: A Document Processing System* or *The L^AT_EX Companion*.

Further information

More information about L^AT_EX 2_ε is to be found in:

L^AT_EX: A Document Preparation System, Leslie Lamport, Addison Wesley, 2nd ed, 1994.

The L^AT_EX Companion, Goossens, Mittelbach and Samarin, Addison Wesley, 1994.

The L^AT_EX distribution comes with documentation on the new features of L^AT_EX:

L^AT_EX 2_ε for authors, describes the new features of L^AT_EX documents, in the file `usrguide.tex`.

L^AT_EX 2_ε for class and package writers, describes the new features of L^AT_EX classes and packages, in the file `clsguide.tex`.

L^AT_EX 2_ε font selection, describes the new features of L^AT_EX fonts for class and package writers, in the file `fntguide.tex`.

For more information on T_EX and L^AT_EX, get in touch with your local T_EX Users Group, or the international T_EX Users Group, P. O. Box 869, Santa Barbara, CA 93102-0869, USA, Fax: +1 805 963 8358, EMail: tug@tug.org.

L^AT_EX News

Issue 2, December 1994

Welcome to L^AT_EX News 2

An issue of *L^AT_EX News* will accompany every future release of L^AT_EX. It will tell you about important events, such as major bug fixes, newly available packages, or any other L^AT_EX news.

December 1994 release of L^AT_EX

December 1994 sees the second release of L^AT_EX 2_ε. We are on schedule to deliver a release of L^AT_EX every six months, in December and June.

This release has seen quite a lot of activity, which is not too surprising as it's only been a year since the first test release of L^AT_EX 2_ε. We don't expect so much activity in the next six months.

Many of the changes are minor improvements and bug-fixes—see *L^AT_EX 2_ε for authors* (`usrguide.tex`), *L^AT_EX 2_ε font selection* (`fntguide.tex`) and our change log (`changes.txt`) for more details.

However, there are two important new packages available for L^AT_EX: `inputenc` and AMS-L^AT_EX.

Accented input

One of the problems with writing non-English documents in L^AT_EX is the accent commands. Reading documents containing text like `na\ "i ve` is frustrating, especially if your keyboard allows you to type `naïve`.

In the past, L^AT_EX has not supported input containing accented characters such as `ï`, because Windows, Macintosh and Unix all have different ways of dealing with accented input, called *input encodings*.

However, the `inputenc` package allows you to specify which input encoding your document is written with, for example to use the ISO Latin-1 encoding, you type:

```
\usepackage[latin1]{inputenc}
```

At the moment, `inputenc` supports the `ascii` and `latin1` input encodings, but more will be added with future releases.

The `inputenc` package is currently a test release. The user interface for the full release will be upwardly compatible with the test version.

AMS-L^AT_EX

AMS-L^AT_EX is a set of miscellaneous extensions for L^AT_EX distributed by the American Mathematical Society. They provide superior information structure

and superior printed output for mathematical documents.

There are far too many features of AMS-L^AT_EX to list here. AMS-L^AT_EX is described in the accompanying documentation, and in *The L^AT_EX Companion*.

Version 1.2beta of AMS-L^AT_EX was released for testing by intrepid users in October 1994. The full release of AMS-L^AT_EX 1.2 is expected in early January 1995.

It will be divided into two bundles:

- the `amsfonts` packages, which give access to hundreds of new mathematical symbols, and new math fonts such as blackboard bold and fraktur.
- the `amsmath` packages, which provide finer control over mathematical typesetting, such as multi-line subscripts, enhanced theorem and proof environments, and improved displayed equations,

For compatibility with older documents, an `amstex` package will be provided.

L^AT_EX on the internet

L^AT_EX has its own home page on the World Wide Web, with the URL:

<http://www.tex.ac.uk/CTAN/latex/>

This page describes L^AT_EX and the L^AT_EX3 project, and contains pointers to other L^AT_EX resources, such as the user guides, the T_EX Frequently Asked Questions, and the L^AT_EX bugs database.

The electronic home of anything T_EX-related is the Comprehensive T_EX Archive Network (CTAN). This is a network of cooperating ftp sites, with over a gigabyte of T_EX material:

```
ftp://ftp.tex.ac.uk/tex-archive/  
ftp://ftp.shsu.edu/tex-archive/  
ftp://ftp.dante.de/tex-archive/
```

For more information, see the L^AT_EX home page.

Further information

For more information on T_EX and L^AT_EX, get in touch with your local T_EX Users Group, or the international T_EX Users Group, P. O. Box 869, Santa Barbara, CA 93102-0869, USA, Fax: +1 805 963 8358, EMail: tug@tug.org.

L^AT_EX News

Issue 3, June 1995

Welcome to L^AT_EX News 3

An issue of *L^AT_EX News* will accompany every future release of L^AT_EX. It will tell you about important events, such as major bug fixes, newly available packages, or any other L^AT_EX news.

June 1995 release of L^AT_EX

June 1995 sees the third release of L^AT_EX 2_ε. We are on schedule to deliver a release of L^AT_EX every six months, in December and June.

In the last *L^AT_EX News*, we said “we don’t expect so much activity in the next six months,” which has turned out not to be true!

Additional input encodings

In the last release of L^AT_EX we distributed a test version of the `inputenc` package which allows the use of input characters other than just a–z and A–Z. The package has proved to be robust, so we are now distributing an expanded version. The new release comes with a number of input encodings:

- `ascii` the standard encoding,
- `latin1` the ISO Western European alphabet,
- `latin2` the ISO Eastern European alphabet,
- `cp437` the IBM codepage 437,
- `cp850` the IBM codepage 850, and
- `applemac` the Apple Macintosh encoding.

These can be used by specifying an option to the `inputenc` package, for example:

```
\usepackage[latin1]{inputenc}
```

The new input encodings are currently being tested, but we don’t expect any major changes.

L^AT_EX getting smaller

In the past releases of L^AT_EX 2_ε, the amount of memory L^AT_EX requires has increased, but we are pleased to say that this trend has been reversed. We hope that future releases of L^AT_EX will continue to get smaller.

For example, on this document, the December 1994 release used 52,622 words of memory, and the June 1995 release uses 51,216 words of memory, which is a 2.7% reduction.

We are currently experimenting with other ways of reducing the size of L^AT_EX. For example, we are

experimenting with an option to remove the `picture` and `tabbing` environments from the L^AT_EX kernel, and to load them from a file the first time they are used. This should help L^AT_EX to run on machines with limited memory. See `autoload.txt` for details.

Distribution and modification

One topic of discussion that has kept us busy is the distribution and modification conditions of L^AT_EX. We are committed to keeping L^AT_EX as free reliable software, and ensuring that (as far as possible) L^AT_EX documents will produce the same results on all systems.

The modification conditions are currently under discussion, and we would like to hear from anyone interested. Please read `modguide.tex` for more information.

AMS-L^AT_EX full release

The AMS-L^AT_EX packages were still in beta test in the December 1994 release of L^AT_EX, and the full release came out in January 1995.

AMS-L^AT_EX is described in the *User’s Guide* (`amsl.doc.tex`) and in *The L^AT_EX Companion*.

PostScript fonts

There is a new test release of the PSNFSS packages for accessing PostScript fonts in L^AT_EX 2_ε. This includes an update to all of the fonts, to remove many of the underfull and overfull `\hbox` warnings, and improve the setting of non-English languages.

The new release of L^AT_EX removes all of the ‘hidden’ uses of Computer Modern mathematics. For example, the footnote markers used to use math mode, so always used Computer Modern digits rather than ones from the current text font. This has now been fixed.

Further information

For more information on T_EX and L^AT_EX, get in touch with your local T_EX Users Group, or the international T_EX Users Group, P. O. Box 869, Santa Barbara, CA 93102-0869, USA, Fax: +1 805 963 8358, EMail: tug@tug.org.

The L^AT_EX home page is <http://www.tex.ac.uk/ctan/latex/> and contains links to other WWW resources for L^AT_EX.

L^AT_EX News

Issue 4, December 1995

Welcome to L^AT_EX News 4

An issue of *L^AT_EX News* will accompany every future release of L^AT_EX. It will tell you about important events, such as major bug fixes, newly available packages, or any other L^AT_EX news. This issue accompanies the fourth release of L^AT_EX 2_ε.

L^AT_EX getting smaller

The last release in June started a trend of L^AT_EX becoming smaller, we are pleased to announce that this has continued with this release. In particular the experimental ‘autoload’ version described in `autoload.txt` is much smaller as more parts of L^AT_EX are autoloaded.

New ‘concurrent’ docstrip

The time taken to ‘unpack’ this release from the documented sources should be much reduced (roughly half the time, depending on installation conditions). This is due to an improved version of the docstrip program that has been contributed by Marcin Woliński. This can write up to 16 files at once. The previous version could only write one file at a time which meant that it was very slow when producing many small files from the same source file as the source needed to be re-read for each file written.

New T1 encoded fonts

This year Jörg Knappen has completed a new release of the ‘Cork’ (T1) encoded Computer Modern fonts: the dc fonts release 1.2.

This release of the dc fonts fixes many bugs (including the missing ? ‘(i) and ! ‘(i) ligatures) and improves the fonts in many other ways. It is strongly recommended that you upgrade as soon as possible if currently you are using the old dc fonts, release 1.1 or earlier. The new fonts are available from the CTAN archives, in `tex-archive/fonts/dc`.

The names of the font files are *different*. This does not affect L^AT_EX documents but *does* affect the installation procedure as it assumes that you have the *new* fonts, and will write suitable ‘fd’ files for those fonts. If you have not yet upgraded your dc fonts then, after unpacking the distribution, you *must* `latex olddc.ins` to produce ‘fd’ files for the old dc fonts. This must be done *before* the format is made. Running the test document at `ltxcheck.tex` the end of

the installation will inform you if the wrong set of ‘fd’ files has been installed.

Note that this change does not affect the standard ‘OT1’ Computer Modern fonts that L^AT_EX uses by default.

More robust commands

The commands `\cite` and `\sqrt` are now robust.

Although most commands with optional arguments are fragile, as documented, such commands defined using the second optional argument of `\newcommand` and its derivatives are now *robust*.

New Interface to building ‘extension’ classes

The mechanism provided by `\DeclareOption`, `\ProcessOptions` and `\LoadClass` has proved to be a powerful and expressive means of defining one class in terms of another ‘base’ class. However there have been some requests to simplify the declaration of the common case where you want the ‘base’ class to be called with *all* the options that were specified to the extension class. This is now provided by the new command `\LoadClassWithOptions`. A similar command `\RequirePackageWithOptions` is provided for package use. More details of this feature are provided in `clsguide.tex` and `ltclass.dtx`.

More Input Encodings

The experimental `inputenc` package allows a more natural style of input of accented and other characters.

Three new input encodings are now supported.

- **ansinew** the Windows ansi encoding, as used in Microsoft Windows 3.x.
- **cp437de** a variant of `cp437`, which uses β rather than β in the appropriate slot.
- **next** the encoding used on Next computers.

Further information

For more information on T_EX and L^AT_EX, get in touch with your local T_EX Users Group, or the international T_EX Users Group, 1850 Union Street, #1637, San Francisco, CA 94123, USA, Fax: +1 415 982 8559, Email: tug@tug.org. The L^AT_EX home page is <http://www.tex.ac.uk/ctan/latex/> and contains links to other WWW resources for L^AT_EX.

L^AT_EX News

Issue 5, June 1996

Welcome to L^AT_EX News 5

This issue of *L^AT_EX News* accompanies the fifth release of the new standard L^AT_EX, L^AT_EX 2_ε.

Extra possibilities for section headings

Most L^AT_EX sectioning commands are defined using `\@startsection`. For example, the `article` class defines:

```
\newcommand\section{\@startsection
{section}{1}{0pt}{-3.5ex plus-1ex minus-.2ex}%
{2.3ex plus.2ex}{\normalfont\Large\bfseries}}
```

The last argument specifies the style in which the section heading is to be typeset.

The new feature added at this release is that at the *end* of this argument you may specify a command that *takes an argument*. This command will be applied to the section number and heading. For example, one could use the `\MakeUppercase` command to produce uppercase headings. A package or class file could contain:

```
\renewcommand\section{\@startsection
{section}{1}{0pt}{-3.5ex plus-1ex minus-.2ex}%
{2.3ex plus.2ex}{\normalfont\Large\MakeUppercase}}
```

to produce section headings using uppercase medium weight text, rather than the bold text used by `article`. Note that, like the font choice, the uppercasing applies only to the actual heading (including any automatically generated section number), not to the text as it may appear in the running head or table of contents.

The ‘openany’ option in the ‘book’ class

The `openany` option allows chapter and similar openings to occur on left hand pages. Previously this option only affected `\chapter` and `\backmatter`. It now also affects `\part`, `\frontmatter` and `\mainmatter`.

More font (output) encodings

The font encoding name T3 has been allocated to the encoding used in the new 256-character IPA fonts (for the phonetic alphabet) produced by Rei Fukui. His package, `tipa`, gives access to these fonts and should soon be available. (The encoding named OT3 is the 128-character encoding used in the IPA fonts produced by Washington State University.)

More input encodings supported

The `inputenc` package now supports the IBM codepage 852 used in Eastern Europe, with the option `[cp852]` contributed by Petr Sojka.

Also, the `inputenc` package now activates most ‘control codes’ with ASCII values below 32. Currently none of the encodings in the standard distribution makes use of these positions.

Fixes and improvements

The L^AT_EX kernel has only had minor changes, apart from `\@startsection` mentioned above. However, some small fixes have been incorporated removing the following problems:

- In `tabular` and `array`, previous versions of L^AT_EX ‘lost’ the inter-column space from an ‘1’-column, when that column was completely empty.
- Previously, the use of the `\nofiles` command could change the *vertical spacing* in a document. A side effect of fixing this is that when `\nofiles` is used, `\label` puts a blank line in the log file.
- L^AT_EX often loads fonts ‘on demand’. Previously, this could happen inside the argument of an accent command and this would cause the accent to appear in the wrong place.

Changes to the ‘tools’ packages

- The `longtable` package now uses a modified algorithm, contributed by David Kastrup, to align the ‘chunks’ of a table. It is now unnecessary to edit the document to add `\setlongtables` before the final run of L^AT_EX. In certain cases of overlapping `\multicolumn` entries, the new algorithm will produce better column widths than the old (at the price of extra passes through L^AT_EX).
- The `dcolumn` package now has the extra possibility of specifying the number of digits both *before* and *after* the ‘decimal point’. This makes it easy to centre the column of numbers under a wide heading.

New copy of the L^AT_EX bug database

<http://www.tex.ac.uk/ctan/latex/bugs.html> will soon have links to a copy of the searchable L^AT_EX bugs database at Mainz (Germany) as well as the original copy at Sussex (England).

L^AT_EX News

Issue 6, December 1996

Welcome to L^AT_EX News 6

This issue of *L^AT_EX News* accompanies the sixth release of the new standard L^AT_EX, L^AT_EX 2_ε.

Mono-case file names

Previously L^AT_EX has used some files with ‘mixed-case’ file names such as `T1cmr.fd` and `T1enc.def`.

These file names cause problems on some systems (in particular they are illegal on the ISO 9660 CDROM format) and so in this release all file names have been made lowercase (for example `t1cmr.fd` and `t1enc.def`).

This change should *not* affect any document. Within L^AT_EX, encodings still have the usual uppercase names in uses such as `\usepackage[T1]{fontenc}` and `\fontencoding{T1}`. L^AT_EX will automatically convert to the lowercase form while constructing the file name. L^AT_EX will input the ‘fd’ file under the old name if it fails to find the file with the new name, so existing collections of fd files should still work with this new release.

The change *does* affect the configuration files that may be used to make the L^AT_EX format with `initex`. For example, the file `fonttext.ltx` previously specified `\input{T1cmr.fd}`. It now has `\input{t1cmr.fd}`. If you use a local file `fonttext.cfg` you will need to make similar changes, as `\input{T1cmr.fd}` will not work as `T1cmr.fd` is no longer in the distribution.

The files affected by this change all have names of the form `*.fd` or `*enc.def`.

Another input encoding

Thanks to work by Søren Sandmann, the `inputenc` package now supports the IBM codepage 865 used in Scandinavia.

Better user-defined math display environments

Suppose that you want to define an environment for displaying text that is numbered as an equation. A straightforward way to do this is as follows:

```
\newenvironment{texeqn}
{\begin{equation}
  \begin{minipage}{0.9\linewidth}}
{\end{minipage}
\end{equation}}
```

However, if you have tried this then you will probably have noticed that it does not work perfectly when used

in the middle of a paragraph because an inter-word space appears at the beginning of the first line after the environment.

There is now an extra command (with a very long name) available that you can use to avoid this problem; it should be inserted as shown here:

```
\newenvironment{texeqn}
{\begin{equation}
  \begin{minipage}{0.9\linewidth}}
{\end{minipage}
\end{equation}
\ignorespacesafterend}
```

Docstrip improvements

The `docstrip` program that is used to unpack the L^AT_EX sources has undergone further development. The new version should be able to process all old ‘batchfiles’ but it allows a simpler syntax in new ‘batchfiles’ (no need to define `\def\batchfile{...}`).

It also allows ‘target’ directories to be specified when writing files. This directory support is disabled by default unless activated in a local `docstrip.cfg` configuration file. See `docstrip.dtx` for details.

AMS L^AT_EX update

Since the last L^AT_EX release in June, the American Mathematical Society have re-issued the ‘AMS L^AT_EX’ classes and packages, fixing several reported problems.

Graphics package update

The L^AT_EX color and graphics packages have been updated slightly, principally to support more dvi drivers, see the `readme` file in the `graphics` distribution.

EC Fonts released

The first release of the Extended Computer Modern fonts has just been made. (In January 1997.)

This release of L^AT_EX does *not* default to these ‘ec’ fonts as its T1 encoded fonts. By default it will use the ‘dc’ fonts if the T1 encoding is requested.

As noted in `install.txt` you may run T_EX on the install file `ec.ins` *after* unpacking the base distribution but *before* making the L^AT_EX format. This will produce suitable ‘fd’ files making L^AT_EX (including, for the first time, the `slides` class) use the ‘ec’ fonts as the default T1 encoded font set.

L^AT_EX News

Issue 7, June 1997

T1 encoded Computer Modern fonts

As in the last release the base L^AT_EX distribution contains three different sets of ‘fd’ files for T1 encoded fonts.

In this release the default installation uses `ec.ins` and so installs files suitable for the current ‘EC fonts’ distribution. If you have still not updated to the EC fonts and are using the earlier test versions, known as DC then you should unpack `newdc.ins` (for DC release 1.2 or later) or `olddc.ins` (for the original releases of the DC fonts). This should be done after unpacking `unpack.ins` but before making the format by running `initEX` on `latex.ltx`. There are further details in `install.txt`.

T1 encoded Concrete fonts

The Metafont sources for T1 encoded ‘Concrete’ fonts have been removed from the `mfns` distribution as they were based on the now obsolete DC fonts release 1.1. Similarly the `cmextra.ins` install file in the base distribution no longer generates fd files for the ‘Concrete’ fonts. To use these fonts in either T1 or OT1 encoding it is recommended that you obtain Walter Schmidt’s `ccfonts` package and fonts from CTAN `macros/latex/contrib/supported/ccfonts`.

Further input encodings

Two more `inputenc` packages have been added: for `latin5`, thanks to H. Turgut Uyar; and for `latin3`, thanks to Jörg Knappen.

Normalising spacing after punctuation

The command `\normalsfcodes` was introduced at the last patch release. This is normally given the correct definition automatically and so need not be explicitly set. It is used to correct a problem, reported by Donald Arseneau, that punctuation in page headers has always (in all known T_EX formats) been potentially incorrect if the page break happens while a local setting of the space codes (for instance by the command `\frenchspacing`) is in effect. A common example of this happening in L^AT_EX is in the `verbatim` environment.

Accessing Bold Math Symbols

The tools distribution contains a new package, `bm`, which defines a command `\bm` that allows individual bold symbols to be accessed within a math expression

(in contrast to `\boldmath` which makes whole math expressions default to bold fonts). It is more general than the existing `amsbsy` package; however, to ease the translation of documents between these two packages, `bm` makes `\boldsymbol` an alias for `\bm`.

This package was previously made available from the ‘contrib’ area of the CTAN archives, and as part of Y&Y’s L^AT_EX support for the MathTime fonts.

Policy on standard classes

Many of the problem reports we receive concerning the standard classes are not concerned with bugs but are suggesting, more or less politely, that the design decisions embodied in them are ‘not optimal’ and asking us to modify them.

There are several reasons why we have decided not to make such changes to these files.

- However misguided, the current behaviour is clearly what was intended when these classes were designed.
- It is not good practice to change such aspects of ‘standard classes’ because many people will be relying on them.

We have therefore decided not to even consider making such modifications, nor to spend time justifying that decision. This does not mean that we do not agree that there are many deficiencies in the design of these classes, but we have many tasks with higher priority than continually explaining why the standard classes for L^AT_EX cannot be changed.

We would, of course, welcome the production of better classes, or of packages that can be used to enhance these classes.

New addresses for TUG

For information about joining the T_EX Users Group, and about lots of other L^AT_EX-related matters, please contact them at their new address:

T_EX Users Group, P.O. Box 1239,
Three Rivers, CA 93271-1239, USA
Fax: +1 209 561 4584
E-mail: tug@mail.tug.org
URL: <http://www.tug.org/>

L^AT_EX News

Issue 8, December 1997

New supported font encodings

Two new font encodings are supported as options to the `fontenc` package:

OT4 This is a seven-bit encoding designed for Polish. The L^AT_EX support was developed by Mariusz Olko.

TS1 This is the ‘Text Companion Encoding’; it contains symbols designed to be used in text, as opposed to mathematical formulas, and some accents designed for uppercase letters. It is currently supported by the ‘tc’ fonts, which match the T1 encoded ‘ec’ text fonts. A subset of the glyphs in this encoding is supported by virtual fonts distributed with the PostScript font metrics on the CTAN archives. (This is the ‘8c’ encoding in Karl Berry’s fontname scheme.) The `textcomp` package provides access to this encoding but here is a warning to current users of that package: some of the internal names for the characters have changed.

New input encodings

These additions to the `inputenc` package are `decmulti` (the DEC Multinational Character Set, contributed by M. Y. Chartoire) and `cp1250` (an MS-Windows encoding for Central and Eastern Europe, contributed by Marcin Woliński). There is also a `cp1252` encoding that is identical to `ansinew`.

Tools

The `calc` package (used in many examples in *The L^AT_EX Companion*) has been contributed to this distribution by Kresten Krab Thorup and Frank Jensen. This is essentially the same as the version that has been available from the CTAN archives for some time, with one minor change: to use L^AT_EX-style error messages. It enables the use of arithmetic expressions within arguments to standard L^AT_EX commands where a length or a counter value is required. For example:

```
\setcounter {page} { \value{page} * 2 + 1 }
\parbox { 3in - ( 2mm + \textwidth / 9 ) }
```

There have also been some improvements to several other packages in this collection. In particular, `bm` now works correctly with constructions such as `\bm{f’}` involving ‘ or other characters which use T_EX’s special “`\mathcode"8000`” feature. Also, `multicol` sets the length `\columnwidth` to an appropriate value; this enables it to work with classes that support two-column setting, e.g., the AMS classes.

Graphics

The special `oztex.def` driver file has been removed, and OzT_EX support has been merged with dvips, following advice from Andrew Trevorrow about OzT_EX 3.x.

The `keyval` package has had some internal improvements: to use L^AT_EX format error messages; and to avoid ‘# doubling’. This latter change means that the `command` key for the `graphicx` version of `\includegraphics` should now be used with one # rather than two. For example, `command = ‘gunzip #1`. Fortunately this key is almost never used in practice, so few if any documents should be affected by this change.

L^AT_EX3 experimental programming conventions

As announced at the T_EX Users Group meeting (Summer 1997), a group of highly experimental packages will soon be released to allow experienced T_EX programmers to experiment with, and comment on, a proposed set of syntax conventions and basic data-types that might form the basis for programming large scale projects in T_EX. They will be located in this CTAN directory:

`CTAN:macros/latex/packages/exp13`

The documentation of this material is as follows: individual package files provide outline, draft documentation; there is an article that gives an overview of the syntax and related concepts; there is a `readme.txt` file containing a brief description of the collection.

All aspects of these packages are liable, indeed likely, to change. They should not be used at this stage for anything that requires a stable system. However, we do encourage people to experiment with these packages, and to send comments on them to the L^AT_EX-L mailing list. To subscribe to this list, mail to:

`listserv@urz.uni-heidelberg.de`

the following one line message:

`subscribe LATEX-L <first-name> <second-name>`

L^AT_EX News

Issue 9, June 1998

New math font encodings

A joint working group of the T_EX Users Group and the L^AT_EX3 Project is developing a new 8-bit math font encoding for T_EX. It is designed to overcome several limitations and implementation problems of the old math font encodings and to simplify switching between different sets of math fonts, much as the L^AT_EX font selection interface has simplified switching between text fonts.

Since the work on this project relies entirely on volunteer work, we cannot give a specific release date yet. However, a prototype implementation already exists. This contains several sets of virtual fonts, some L^AT_EX packages and a kernel module; we hope to integrate it into the main L^AT_EX distribution for the next release.

Documents using only standard L^AT_EX commands for math symbols should not be affected by switching to the new math font encodings. However, documents, classes or packages making specific assumptions about the encoding of math symbol fonts are likely to break.

Further information about the Math Font Group may be found on the World Wide Web at <http://www.tug.org/twg/mfg/>.

A new math accent

A new math accent, `\mathring`, has been added. This is a math mode version of the ring accent (°) which is available in text mode with the command `\r`.

Extended `\DeclareMathDelimiter`

The command `\DeclareMathDelimiter` has been extended. Normally this command takes six arguments. Previously, when being used to declare a character (such as `[]`) as a delimiter, a variant form was used with only five arguments. The argument specifying the default ‘math class’ was omitted. Now the full six-argument form may be used in this case. The extra information is used to implicitly declare the character via `\DeclareMathSymbol` for use when the symbol is not used with `\left` or `\right`.

The old five-argument form is detected and will work as before.

Tools distribution

The `multicol` package now supports the production of multiple columns without balancing the last page. To get this effect use the `multicols*` environment.

The `layout` package was partly recoded by Hideo Umeki to display page layout effects in a better way.

As suggested by Donald Arseneau, the `calc` package was extended to support the new commands `\widthof{text}`, `\heightof{text}`, and `\depthof{text}` within a `calc`-expression. At the same time we modified a few kernel commands so that `calc`-expressions can now be used in various useful places such as the dimension arguments to the `tabular` environment and the `\rule` command. For many other standard L^AT_EX commands this was already possible.

Support for Cyrillic encodings

We are very pleased that, after a lengthy period of development, a set of fonts, encodings and support files for using L^AT_EX with Cyrillic characters will soon be available.

Test versions of the ‘LH’ fonts for these Cyrillic encodings, based on the Computer Modern design, are available from CTAN archives in the directory `fonts/cyrillic/lh-test`. The L^AT_EX support files (by Werner Lemberg and Vladimir Volovich) are also available from CTAN archives in `macros/latex/contrib/supported/t2`

Default docstrip header

Many L^AT_EX users now distribute packages in documented source form using the `docstrip` system. Docstrip allows a header to be placed on generated package files, suitable for giving copyright information, or distribution conditions.

We have changed the default version of this header so that it allows stripped files to be distributed in ready-to-run installations such as the T_EXLive CD. If you use the default header for distributing your files you should check that the new copyright text is acceptable to you. The file `docstrip.dtx` explains how to produce your own header if you wish to do so.

L^AT_EX News

Issue 10, December 1998

Five years of L^AT_EX 2_ε

Since this is the 10th edition of L^AT_EX News, the (no longer) New Standard L^AT_EX must have hit the streets almost this long ago. In fact it was only the beta-version that some people got just in time for Christmas 1993, and since then there has been a lot of tidying-up and smoothing of rough edges (not to mention a few bug fixes!).

Maybe it is time for something more radically different to emerge and be hungrily adopted by the world; but don't panic, we shall be maintaining what you have now for a long time yet. Amongst the more polite things that have been written about our efforts, we found that this quote (somewhat censored to protect the guilty) well reflects some of our feelings about working on L^AT_EX over the years: *the mere existence of L^AT_EX 2_ε is a great miracle.*

Restructuring the L^AT_EX distribution

Since the (once) 'new' standard L^AT_EX has reached such a venerable age, we are reviewing the way in which the system is presented to the world.

An early intention is to define, given the wide variety of good packages now available, what now constitutes a useful installation of L^AT_EX. We also hope that such a definition will help document portability if it leads to a future in which a L^AT_EX class designer can reasonably assume that a known list of facilities will be there for all users (so that each class need not supply them).

As a first small step towards this definition, we shall replace the `latex/packages` subdirectory on CTAN. This directory was a curious mixture of the important, such as the L^AT_EX `tools`, that any self-respecting L^AT_EX installation ought to have, and the esoteric or experimental.

The esoterica from `packages` will be moved to new locations, as follows:

```
expl3 to latex/exptl/project
mfnfss to latex/contrib/supported/mfnfss
```

The subdirectory that replaces `packages` will be called `latex/required`; all the other sub-directories of `packages` will be moved there.

L^AT_EX Project on the Internet

A new `latex-project.org` domain has been registered. The web site is not yet fully functional but the old L^AT_EX pages from CTAN are available at <http://www.latex-project.org/> and the L^AT_EX bug reporting address has been changed to `latex-bugs@latex-project.org`.

Restructuring the L^AT_EX package licenses

Several people have requested an easy mechanism for the distribution of L^AT_EX packages and other software "under the same conditions as L^AT_EX". The old `legal.txt` file was unsuitable as a general licence as it referred to specific L^AT_EX authors, and to specific files.

Therefore, in this release `legal.txt` contains just the copyright notice and a reference to the new *L^AT_EX Project Public License* (LPPL) for the distribution and modification conditions. The `tools`, `graphics`, and `mfnfss` packages also now refer to this license in their distribution notices.

Support for Cyrillic encodings

Basic Cyrillic support, as announced in L^AT_EX News 9, is now finally an official part of L^AT_EX. It includes support for the following standard Cyrillic font encodings (this list may grow): T2A T2B T2C X2.

It also includes various Cyrillic input encodings (20 in total, including commonly used variants and Mongolian Cyrillic encodings). This provides platform independent and sophisticated basic support for high-quality typesetting in various Cyrillic-based languages.

For further information see the file `cyrguide.tex`.

Tools distribution

The `varioref` package has been extended to support textual page references to a range of objects: e.g., if `eq-first` and `eq-last` are the label names for the first and last equation in a sequence, then you can now write

```
see~\vrefrange{eq-first}{eq-last}
```

This results in different text depending on whether both labels fall on the same page.

Some additional user commands, as well as building-blocks for writing private extensions, are described in the accompanying documentation.

L^AT_EX News

Issue 11, June 1999

Back in sync

The last release of L^AT_EX was delayed even longer than you have come to expect. We hope that it proved worth waiting for. It required a major integration of the code from several people and, independently, the introduction of the LPPL (see L^AT_EX News 10) plus several related changes to our internal systems. It therefore seemed sensible to wait until everything was complete rather than do things in too much hurry.

This seem to have been a successful strategy as the recent patch release was related to an isolated change that was done many months previously. If this release does not appear a lot closer to its nominal date then ... well, you will not be reading this sentence!

Yearly release cycles

With the year 2000 rapidly approaching, we intend to switch to a release frequency of just one per year (with patches if necessary) for the core of L^AT_EX 2_ε. These days the system is sufficiently stable that the original update policy is costing everybody more time than is now warranted.

LPPL update

Thanks to extensive and valuable input from Matt Swift (swift@alum.mit.edu) we now have a clearer and more detailed form of the L^AT_EX Project Public Licence. This release contains both the original version (in `lppl-1-0.txt`) and the updated version, LPPL 1.1.

The future of SliT_EX

We still get a very small trickle of reports about this part of the system (if you are no longer able to recall L^AT_EX 2.09 then you will know it as the `slides` class). We have not classified them (in our minds at least) as bugs since we have always known that there are many problems with this class. It is clear to us that the only sensible action would be to redesign the system completely; in particular, to remove much of its complexity whose purpose is to support 10-year-old overlay technology. However, this would take a lot too much time and would be completely out of proportion to its current usage.

We are therefore planning to make the `slides` class unsupported in the sense that any problem related to the use of invisible fonts is considered to be a feature (The L^AT_EX 2_ε manual by Leslie Lamport doesn't even

describe this part of the class any more). Of course, if it still has its enthusiasts then we are happy to cede it to their loving care (somewhat like a preserved steam locomotive, in some parts of the world).

Fontenc package peculiarities

The `\usepackage` interface normally ensures that a package is loaded only once. The `fontenc` package has become an exception to this rule: it can be loaded several times using different options, e.g., allowing the user to add a font encoding in the preamble. This comes at a price for package writers: the low-level commands (see `ltclass.dtx`) used to check if a package was loaded, and with which options, do not work for the `fontenc` package.

New math font encodings

As we announced in L^AT_EX News 9, a joint working group of the T_EX Users Group and the L^AT_EX3 Project has developed a new 8-bit math font encoding for T_EX. The reason why this work is not yet released is because of other exciting developments in the world of math fonts and math characters. It is obviously wise to ensure that the encoding work is fully integrated with the available fonts.

Those interested are reminded that further information about the Math Font Group may be found on the World Wide Web at:

<http://www.tug.org/twg/mfg/>.

Tools distribution

The `multicol` package has now got a small but useful extension which allows you to force a column break where this is really necessary. This is done with the command `\columnbreak`, which can be used like `\pagebreak` (e.g., within paragraphs) except that it cannot have an optional argument and thus it always forces a new column.

Coming soon

Major work on a new class file structure to support flexible designs is well under way; some of this work will be presented at the TUG'99 conference in Vancouver, Canada. With a bit of luck much of this work could be ready for integration into the next release—so watch this space!

L^AT_EX News

Issue 12, December 1999

LPPL update

Since the release of the L^AT_EX Project Public Licence version 1.1, we have received a small number of queries which resulted in some minor changes to improve the wording or explain the intentions better. As a consequence this release now contains LPPL 1.2 in the file `lppl.txt` and the previous versions as `lppl-1-0.txt` and `lppl-1-1.txt`.

fixltx2e package

This package provides fixes to L^AT_EX 2_ε which are desirable but cannot be integrated into the L^AT_EX 2_ε kernel directly as they would produce a version incompatible to earlier releases (either in formatting or functionality).

By having these fixes in the form of a package, users can benefit from them without the danger that their documents will fail, or produce unexpected results, at other sites; this works because a document will contain a clear indication (the `\usepackage` line, preferably with a required date) that at least some of these fixes are required to format it.

Outcome of TUG '99 (Vancouver)

The slides from the TUG'99 presentation we gave on *a new interface for L^AT_EX class designers* are available from the L^AT_EX Project website; look for the file `tug99.pdf` at:

<http://www.latex-project.org/talks/>

Please note that this document was intended only to be informal “speaker’s notes” for our own use. We decided to make them available (the speaker’s notes as well as the slides that were presented) because several people requested copies after the talk. However, they are *not* in a polished copy-edited form and are not intended for publication.

Prototype implementations of parts of this interface are now available from:

<http://www.latex-project.org/code/experimental/>

We are continuing to add new material at this location so as to stimulate further discussion of the underlying concepts. As of December 1, 1999 the following parts can be downloaded.

xparse Prototype implementation of the interface for declaring document command syntax. See the `.dtx` files for documentation.

template Prototype implementation of the template interface (needs parts of **xparse**).

The file `template.dtx` in that directory has a large section of documentation at the front describing the commands in the interface and giving a ‘worked example’ building up some templates for caption formatting.

xcontents Interface description for table of contents data (no code yet). Coding examples have been thoroughly discussed on the `latex-l` list.

xfootnote Working examples for generating footnotes, etc. Needs **xparse** and **template**.

All examples are organised in subdirectories and additionally available as **gzip tar** files.

Please remember that this material is intended only for experimentation and comments; thus any aspect of it, e.g., the user interface or the functionality, may change and, in fact, is very likely to change. For this reason it is explicitly forbidden to place this material on CD-ROM distributions or public servers.

These concepts, as well as their implementation, are under discussion on the list `LATEX-L`. You can join this list, which is intended solely for discussing ideas and concepts for future versions of L^AT_EX, by sending mail to `listserv@URZ.UNI-HEIDELBERG.DE` containing the line

SUBSCRIBE LATEX-L *Your Name*

This list is archived and, after subscription, you can retrieve older posts to it by sending mail to the above address, containing a command such as:

GET LATEX-L LOGyy^{mm}

where `yy`=Year and `mm`=Month, e.g.

GET LATEX-L LOG9910

for all messages sent in October 1999.

L^AT_EX News

Issue 13, June 2000

Yearly release cycle

We announced in *L^AT_EX News 11* that we intended to switch to a 12-monthly release schedule. With the present (June 2000) release, this switch is being made: thus the next release of L^AT_EX will be dated June 2001. We shall of course continue, as in the past, to release patches as needed to fix significant bugs.

PSNFSS: Quote of the Month

You should say in the L^AT_EX News that Walter Schmidt has taken over PSNFSS from me. It gives me a certain pleasure to be able to draw a line under that part of my life...

Sebastian Rahtz

The PSNFSS material, which supports the use of common PostScript fonts with L^AT_EX, has been thoroughly updated. Most noticeably, the **mathpple** package, which used to be distributed separately, is now part of the basic PSNFSS bundle; this package provides mathematical typesetting with the Palatino typeface family. In addition, numerous bugs and flaws have been fixed and the distribution has been ‘cleaned up’. The file **changes.txt** contains a detailed list of these changes.

The documentation (in **psnfss2e.pdf**) has been completely rewritten to provide a comprehensive introduction to the use of PostScript fonts.

Notice that the new PSNFSS needs updated files for font metrics, virtual fonts and font definitions. If you received the new version (8.1) as part of a complete T_EX system then these new font files should also have been installed. However, if you intend to install or update PSNFSS yourself, please read the instructions in the file **00readme.txt** of the new PSNFSS distribution.

Support for commercial PostScript fonts, such as Lucida Bright, has been removed from the basic distribution; it is now available from CTAN: <http://mirror.ctan.org/macros/latex/contrib/supported/psnfssx>.

New AMS-L^AT_EX

Version 2.0 of AMS-L^AT_EX was released on December 1, 1999. It can be obtained via <ftp://ftp.ams.org/pub/tex/> or <http://www.ams.org/tex/amslatex.html>, as well from CTAN: <http://mirror.ctan.org/macros/latex/required/amslatex>.

This release consists chiefly of bug fixes and consolidation of the existing features. The division of

AMS-L^AT_EX into two main parts (the math packages; the AMS document classes) has been made more pronounced. The files **diffs-m.txt**, **diffs-c.txt**, **amsmath.faq**, and **amscs.faq** describe the changes and address some common questions.

The primary documentation files remain **amslatex.tex**, for the amsmath package, and **instr-1.tex**, for the AMS document classes. The documentation for the **amsthm** package, however, has been moved from **amslatex.tex** to a separate document **amsthdoc.tex**.

New input encoding latin4

The package **inputenc** has, thanks to Hana Skoumalová, been extended to cover the **latin4** input encoding; this covers Baltic and Scandinavian languages as well as Greenland Inuit and Lappish.

New experimental code

In *L^AT_EX News 12* we announced some ongoing work towards a ‘Designer Interface for L^AT_EX’ and we presented some early results thereof. Since then, at Gutenberg 2000 in Toulouse and TUG 2000 in Oxford, we described a new output routine and an improved method of handling vertical mode material between paragraphs. In combination these support higher quality *automated*¹ page-breaking and page make-up for complex pages—the best yet achieved with T_EX!

A paper describing the new output routine is at <http://www.latex-project.org/papers/xo-pfloat.pdf>. All code examples and documentation are available at <http://www.latex-project.org/code/experimental/>.

This directory has been extended to contain

galley Prototype implementation of the interface for manipulating vertical material in galleys.

xinitials Prototype implementation of the interface for paragraph initials (needs the **galley** package).

xtheorem Contributed example using the **template** package to provide a designer interface for theorem environments.

xoutput A prototype implementation of the new output routine as described in the **xo-pfloat.pdf** paper. Expected availability: at or shortly after the TUG 2000 conference.

¹The stress here is on *automated*!

L^AT_EX News

Issue 14, June 2001

Future releases

We are currently exploring how to best support the very large community of individuals, organisations and enterprises that depend on the robustness and availability of the current standard L^AT_EX distribution. The results of this may lead to some changes in the regular release schedule and the handling of bug reports during the next year.

New release of Babel (required)

Earlier this year a new release of Babel (3.7) became available. You can read about its new features in <http://www.ctan.org/tex-archive/macros/latex/required/babel/announce.txt>

One of the bugs that got fixed in this release deals with how labels are handled by L^AT_EX. Because this part of the kernel is modified by `babel`, the relevant changes need to be coordinated. Therefore to use Babel with this release of L^AT_EX you will need to update your version of `babel` to at least 3.7.

New input encoding latin9

The package `inputenc` has, thanks to Karsten Tinnfeld, been extended to cover the `latin9` input encoding. The ISO-Latin 9 encoding is a useful modern replacement for ISO-Latin 1 that contains a few characters needed for French and Finnish. Of wider interest, it also contains the euro currency sign; this could be the killer argument for many 8-bit texts to use Latin-9 in the future.

According to a Linux manpage, ISO Latin-9 supports Albanian, Basque, Breton, Catalan, Danish, Dutch, English, Estonian, Faroese, Finnish, French, Frisian, Galician, German, Greenlandic, Icelandic, Irish Gaelic, Italian, Latin, Luxemburgish, Norwegian, Portuguese, Rhaeto-Romanic, Scottish Gaelic, Spanish and Swedish. The characters added in `latin9` are (in L^AT_EX notation):

```
\texteuro \v S \v s \v Z \v z \OE \oe \" Y
They displace the following characters from latin1:
\textcurrency \textbrokenbar \"{} \'{} \c{}
\textonequarter \textonehalf \textthreequarters
```

New tools

The new package `trace` provides many commands to control L^AT_EX's tracing and debugging output, including the excellent new information available with ϵ -T_EX such as the extremely useful tracing of local assignments. You will find it in the tools distribution.

It offers the command `\tracelon`, which is similar to `\tracingall` but suppresses uninteresting stuff such as font loading by NFSS (which can go on for pages if you are unlucky). It also offers `\traceoff` to ... guess what! Full details are in the documented source file, `trace.dtx`.

In the base `ifthen` package we have added the uppercase synonyms `\NOT` `\AND` and `\OR`.

New experimental code

In *L^AT_EX News 12* we announced some ongoing work towards a 'Designer Interface for L^AT_EX' and we presented some early results thereof. Since then, at Gutenberg 2000 in Toulouse and TUG 2000 in Oxford, we described a new output routine and an improved method of handling vertical mode material between paragraphs. In combination these support higher quality *automated*¹ page-breaking and page make-up for complex pages—the best yet achieved with T_EX!

More recently we have added material to handle the complex front matter requirements of journal articles; this was presented at Gutenberg 2001 in Metz.

A paper describing the new output routine is at <http://www.latex-project.org/papers/xo-pfloat.pdf>. All code examples and documentation are available at <http://www.latex-project.org/code/experimental>

This directory has been extended to contain the following.

galley Prototype implementation of the interface for manipulating vertical material in galleys.

xinitials Prototype implementation of the interface for paragraph initials (needs the `galley` package).

xtheorem Contributed example using the `template` package to provide a designer interface for theorem environments.

xor A prototype implementation of the new output routine as described in the `xo-pfloat.pdf` paper.

xfrontm A prototype version of the new front matter interface.

¹The stress here is on *automated*!

L^AT_EX News

Issue 15, December 2003

Anniversary release

Yes, it's now 10 years since the first release in this series and, for Knuthists, this release also contains *Issue 16*!

Meanwhile this *Issue 15* describes the major new features in the current release whilst *Issue 16* looks a little way into the future of L^AT_EX.

LPPL – new version

Most importantly, there is now a new version, 1.3, of the L^AT_EX Project Public Licence. Many of you will be thrilled to know that, following the exchange of over 1600 e-mail messages dissecting various aspects of its philosophy such as ‘how many angels can appear in the name of a file before it becomes non-free’, this version is now officially a DFSG (Debian Free Software Guidelines) approved license. The discussions start at <http://lists.debian.org/debian-legal/2002/debian-legal-200207/threads.html> with high traffic throughout August to October 2002 and further heated discussions starting in April 2003 and concluding around June at <http://lists.debian.org/debian-legal/2003/debian-legal-200306/msg00206.html>.

The important features of the new version are useful clarifications in the wording, and revised procedures for making a change to the Current Maintainer of a package. Special thanks to all those people from Debian Legal who worked constructively with us on this onerous task, especially but not exclusively Jeff Licquia and Branden Robinson.

Small updates to `varioref`

The English has been corrected in `\reftextbefore` (an incompatible change). There are other extensions such as `\labelformat`, `\Ref`, `\Vref` and `\vpagerefnum`. Some Dutch text has also been changed and two new options added: `slovak` and `slovene`.

New and more robust commands

Many of the math mode commands for compound symbols have been made robust and a new robust command has been added: `\nobreakdashes`. This last is a low-level command, borrowed from the `amsmath` package, for use only before hyphens or dashes. It prevents the line break that is normally allowed after the following sequence of dashes.

Fixing font sizes

The new `fix-cm` package, by Walter Schmidt, changes the CM font definition (`.fd`) files so that similar design sizes are used in both the OT1 and T1 encodings.

Font encodings

A number of options have been added to the `textcomp` package, enabling only available glyphs to be used. Also, the ‘NFSS font families’ are now divided into five different groups according to the subset of glyphs each provides from the full collection of symbols in the TS1 encoding. Given sufficient information about a font family `textcomp` will use this in order to limit the typesetting to those glyphs that are available.

Use of this mechanism has also enhanced `\oldstylenums` to use the current font if possible.

Displaying font tables

With the `nfssfont` package you can now specify the font to display by giving its ‘NFSS classification’, rather than needing to know its external font file’s name. It is also now possible to generate large collections of font tables in batch mode by providing a suitable input file.

New input encodings

The `inputenc` package has been extended as follows: `macce` input encoding (Apple Central European), thanks to Radek Tryc and Marcin Wolinski; `cp1257` for Baltic languages; `latin10`, thanks to Ionel Ciobîcă. The euro symbol has by now been added to several encodings: `ansinew`, `cp1250` and `cp1252` (which also has another addition), whilst `cp858` adds it to `cp850`.

Unicode input

Partial, experimental support for text files that use the Unicode encoding form UTF-8 is now provided by the option `utf8` for the `inputenc` package.

The only Unicode text file characters supported by the current version are those based on the most common inputs for glyphs from the small collection of standard L^AT_EX Latin encodings.

And finally ... `pict2e`

The old, non-functional version of this package has been removed as there is now a fully working version from Hubert Gäßlein and Rolf Niepraschk. It is described in *The L^AT_EX Manual*.

L^AT_EX News

Issue 16, December 2003

Anniversary news

This anniversary *Issue 16* takes a brief look into the future work of the L^AT_EX3 Project Team, both short and long range. Please let us know if you want to get involved with us in any of this work (see below).

An overview of the 10th Anniversary Release, dated 2003/12/01, is can be found in *Issue 15*.

TLC2: The L^AT_EX Companion – 2nd edition!

Since you are reading this newsletter, there is a good chance that you, or a friend, has already bought this encyclopedic volume: the incomparable Second Edition of this work that is every L^AT_EXie's ultimate lucky charm.

If by some chance you have not yet purchased your own copy then get into training, get shopping, and get flexing your muscles (both physical—it's 1100+ pages, and intellectual) by using it to discover masses of invaluable 'insider information' about:

- the latest release of Standard L^AT_EX;
- over 200 extension packages;
- plus related software and systems.

For more information on this all new (??... OK, not *all*, but over 90%!), all accurate (we hope!) 10th Anniversary Edition, check out <http://www.awprofessional.com/titles/0201362996>.

Future maintenance

We are currently exploring how best to support the very large and rapidly growing community of individuals, organisations and enterprises that depend on the robustness and availability of the current standard L^AT_EX distribution. Although we remain firmly resolved not to make changes in the base distribution (the kernel) of Standard L^AT_EX, there is still much that needs doing to maintain its reliability and utility and to keep up the necessary level of communication with users and supporters. Also, as with all advanced software systems, bugs are still turning up occasionally so some fixes are still essential.

One major impediment to providing adequate service levels in this area is, of course, the difficulties inherent in obtaining the time and commitment of skilled minds—hence the appeal above to anyone interested in getting involved.

LPPL certification

There are still some outstanding diplomatic tasks around the L^AT_EX Project Public Licence: these include e.g., getting it 'OSF certified' and ensuring that it gains more support and wider use, even in the FSF world where it has long been tolerated.

Use of ϵ -T_EX/pdfT_EX

We expect that within the next two years, releases of L^AT_EX will change modestly in order to run best under an extended T_EX engine that contains the ϵ -T_EX primitives, e.g., ϵ -T_EX or pdfT_EX. The details of this possible upgrade need further work so we are not making a definite announcement yet.

Although the current release does not *require* ϵ -T_EX features, we certainly recommend using an extended T_EX, especially if you need to debug macros.

End of 'autoload' support

As computer systems generally grow in capacity, requirements change and so we believe that the *autoload* variant of L^AT_EX is no longer required. Thus, although the code remains it is no longer supported. We hope this does not cause any problems.

New models, new code

In the period 1999–2001 we published many results of our work over the previous decade on the development of new concepts and models for automated typesetting based on T_EX as the underlying platform. These can be found at <http://www.latex-project.org/papers/> and <http://www.latex-project.org/code/experimental/>.

Since then a very large proportion of the The Team's efforts have been diverted to provide the core author team for TLC2, which provides over 1000 pages of carefully researched and tested documentation of many aspects of the vast world of L^AT_EX related software that was developed over that same time period and that continues to grow and improve prodigiously.

Completion of that task ... until TLC3!! ... presents the possibility of getting back to this more exciting development work, or even to more radical work on non-T_EX-based models and implementations.

Of course, any such ideas are predicated on our ability to organise (with you, we hope) an efficient but responsive maintenance and support system for Standard L^AT_EX.

L^AT_EX News

Issue 17, December 2005

Project licence news

The L^AT_EX Project Public License has been updated slightly so that it is now version 1.3c. In the warranty section the phrase “unless required by applicable law” has been reinstated, having got lost at some point. Also, it now contains three clarifications: of the difference between “maintained” and “author-maintained”; of the term “Base Interpreter”; and when clause 6b and 6d shall not apply.

Following requests, we now also provide the text of the licence as a L^AT_EX document (in the file `lpp1.tex`). This file can be processed either as a stand-alone document or it can be included (without any modification) into another L^AT_EX document, e.g., as an appendix, using `\input` or `\include`.

New guide on font encodings

Way back in 1995 work was started on a guide to document the officially allocated L^AT_EX font encoding names. However, for one reason or another this guide (named *L^AT_EX font encodings*) was, until now, not added to the distribution. It describes the major 7-bit and 8-bit font encodings used in the L^AT_EX world and explains the restrictions required of conforming text font encodings. It also lists all the ‘encoding specific commands’ (the LICR or L^AT_EX Internal Character Representation) for characters supported by the encodings OT1 and T1.

When the file `encguide.tex` is processed by L^AT_EX, it will attempt to typeset an encoding table for each encoding it describes. For this to be possible, L^AT_EX must be able to find `.tfm` files for a representative example font for each encoding. If L^AT_EX cannot find such a file then a warning is issued and the corresponding table is omitted.

Robust commands in math

The font changing commands in text-mode have been robust commands for years, but the same has not been true for the math versions such as `\mathbf`. While the math-mode commands worked correctly in section heads, they could cause problems in other places such as index entries. With this release, these math-mode commands are now robust in the same way as their text-mode counterparts.

Updates of required packages

Several of the packages in the tools bundle have been updated for this release.

The `xspace` package has some new features. One is an interface for adding and removing the exceptions it knows about and another is that it works with active characters. These remove problems of incompatibility with the `babel` system.

In *L^AT_EX News 16* we announced that some packages might begin to take advantage of ε -T_EX extensions on systems where these are available: and the latest version of `xspace` does just that. Note also that `fixltx2e` will make use of the facilities in ε -T_EX whenever these are present (see below).

The `calc` package has also been given an update with a few extra commands. The commands `\maxof` and `\minof`, each with two brace-delimited arguments, provide the usual numeric max and min operations. The commands `\settototalheight` and `\totalheightof` work like `\settoheight` and `\heightof`. There are also some internal improvements to make `calc` work with some more primitive T_EX constructs, such as `\ifcase`.

The `varioref` package has acquired a few more default strings but there are still a number of languages for which good strings are still missing.

The `showkeys` package has also been updated slightly to work with more recent developments in `varioref`. Also, it now provides an easy way to define the look of the printed labels with the command `\showkeyslabelformat`.

Work on L^AT_EX fixes

The package known as `fixltx2e` has three new additions. A new command `\textsubscript` has been added as a complement to the command `\textsuperscript` in the kernel. Secondly, a new form of `\DeclareMathSizes` that allows all of its arguments to have a dimension suffix. This means you can now use expressions such as `\DeclareMathSizes{9.5dd}{9.5dd}{7.4dd}{6.6dd}`.

The third new addition is the robust command `\TextOrMath` which takes two arguments and executes one of them when typesetting in text or math mode respectively. This command also takes advantage of ε -T_EX extensions if available; more specifically, when the ε -T_EX extensions are available, it does not destroy kerning between previous letters and the text to be

typeset. The command is also used internally in `fixltx2e` to resolve a problem with `\fnsymbol`.

Also, further work has been done on reimplementing the command `\addpenalty`, which is used internally in several places: we hope it is an improvement!

The graphics bundle

The `graphics` bundle now supports the `dvipdfmx` post-processor and Jonathan Kew's `XETEX` program. By support we mean that the `graphics` packages recognize the new options `xetex` and `dvipdfmx` but we do not distribute the respective driver files.

This leads elegantly to a description of the new policy concerning such driver files in the `graphics` bundle. Most driver files for our `graphics` packages are maintained by the developers of the associated post-processor or `TEX` programs. The teams developing these packages are working very hard: their rapid development offers a stark contrast to the current schedule of `LATEX` releases. It is therefore no longer practical for the `LATEX` Team to be responsible for distributing the latest versions of these driver files.

Therefore the installation files for `graphics` have been split: there is now `graphics.ins` to install the package files and `graphics-drivers.ins` for the driver files (located in `drivers.dtx`). There is no need to install all those provided in the file `drivers.dtx`.

Please also note that, as requested by the maintainers of `PStricks`, we have removed the package `pstcol` as current versions of `PStricks` make it obsolete.

Future development

The title of this section is a little misleading as it actually describes *current* development. In 1998 the `expl3` bundle of packages was put on CTAN to demonstrate a possible `LATEX3` programming environment. These packages have been lying dormant for some time while the `LATEX` Project Team were preoccupied by other things such as developing the experimental packages `xor`, `template`, etc., (and also writing that indispensable and encyclopaedic volume, *The L^AT_EX Companion* – 2nd edition).

In October 2004 work on this code base was resumed with the goal of some day turning it into a kernel for `LATEX3`. This work can now also make full use of the widely accepted ε -`TEX` extensions. Currently two areas are central to this work.

- Extending the kernel code of `LATEX3`.
- Converting the experimental packages such as `xor`, `template` to use the new syntax internally.

Beware! Development of `expl3` is happening so fast that the descriptions above might be out of date when you read this! If you wish to see what's going on then go to <http://www.latex-project.org/code.html> where you can download fully working code (we hope!).

L^AT_EX News

Issue 18, December 2007

This news never existed.

L^AT_EX News

Issue 19, September 2009

New L^AT_EX release

This issue of L^AT_EX News marks the first release of a new version of L^AT_EX 2_ε since the publication of The L^AT_EX Companion in 2005–2006.

Just in time for T_EX Live 2009, this version is a maintenance release and introduces no new features. A number of small changes have been made to correct minor bugs in the kernel, slightly extend the Unicode support, and improve various aspects of some of the `tools` packages.

New code repository

Since the last L^AT_EX release, the entire code base has been moved to a public SVN repository¹ and the entire build architecture re-written. In fact, it has only been possible for us to consider a new L^AT_EX release since earlier this year when the test suite was finally set up with the new system. In the process, a bug in the L^AT_EX picture fonts distributed with T_EX Live was discovered, proving that the tests are working and are still very valuable.

Now that we can easily generate new packaged versions of the L^AT_EX 2_ε distribution, we expect to be able to roll out bug fixes in a much more timely manner than over the last few years. New versions should be distributed yearly with T_EX Live. Having said this, the maintenance of the L^AT_EX 2_ε kernel is slowing down as the bugs become fewer and more subtle. Remember that we cannot change any of the underlying architecture of the kernel or any design decisions of the standard classes because we must preserve backwards compatibility with legacy documents at all costs.

Even new features cannot be added, because any new documents using them will not compile in systems (such as journal production engines) that are generally not updated once they've been proven to work as necessary.

None of this is to say that we consider L^AT_EX 2_ε to be any less relevant for document production than in years past: a stable system is a useful one. Moreover, the package system continues to provide a flourishing and stable means for the development of a wide range of extensions.

Babel

One area of the L^AT_EX 2_ε code base that is still receiving feedback to be incorporated into the main distribution is the Babel system for multilingual typesetting. While the Babel sources have already been added to the SVN repository the integration of the test system for Babel is still outstanding.

The future

While work on L^AT_EX 2_ε tends to maintenance over active development, the L^AT_EX 3 project is seeing new life. Our goals here are to provide a transition from the L^AT_EX 2_ε document processing model to one with a more flexible foundation. Work is continuing in the `expl3` programming language and the `xpackages` for document design. Future announcements about L^AT_EX 3 will be available via the L^AT_EX Project website and in TUGboat.

¹<http://www.latex-project.org/svnroot/latex2e-public/>

L^AT_EX News

Issue 20, June 2011

Scheduled L^AT_EX bug-fix release

This issue of L^AT_EX News marks the first bug-fix release of L^AT_EX 2_ε since shifting to a new build system in 2009. Provided sufficient changes are made each year, we expect to repeat such releases once per year to stay in sync with T_EX Live. Due to the excitement of T_EX's 25th birthday last year, we missed our window of opportunity to do so for 2010. This situation has been rectified this year!

Continued development

The L^AT_EX 2_ε program is no longer being actively developed, as any non-negligible changes now could have dramatic backwards compatibility issues with old documents. Similarly, new features cannot be added to the kernel since any new documents written now would then be incompatible with legacy versions of L^AT_EX.

The situation on the package level is quite different though. While most of us have stopped developing packages for L^AT_EX 2_ε there are many contributing developers that continue to enrich L^AT_EX 2_ε by providing or extending add-on packages with new or better functionality.

However, the L^AT_EX team certainly recognises that there are improvements to be made to the kernel code; over the last few years we have been working on building, expanding, and solidifying the `expl3` programming layer for future L^AT_EX development. We are using `expl3` to build new interfaces for package development and tools for document design. Progress here is continuing.

Release notes

In addition to a few small documentation fixes, the following changes have been made to the L^AT_EX 2_ε code; in accordance with the philosophy of minimising forwards and backwards compatibility problems, most of these will not be noticeable to the regular L^AT_EX user.

Font subsets covered by Latin Modern and T_EX Gyre The Latin Modern and T_EX Gyre fonts are a modern suite of families based on the well-known Computer Modern and ‘PostScript 16’ families with many additional characters for high-quality multilingual typesetting.¹

¹See their respective TUGboat articles for more information:
<http://www.tug.org/TUGboat/tb24-1/jackowski.pdf>

Information about their symbol coverage in the TS1 encoding is now included in `textcomp`'s default font definitions.

Redefinition of `\enddocument` Inside the definition of `\end{document}` the `.aux` file is read back in to resolve cross-references and build the table of contents etc. From 2.09 days this was done using `\input` without any surrounding braces which could lead to some issues in boundary cases, especially if `\input` was redefined by some package. It was therefore changed to use L^AT_EX 2_ε's internal name for this function. As a result, packages that modify `\enddocument` other than through the officially provided hooks may need to get updated.

Small improvement with split footnotes in `ftnright` If in the first column there is more than a full column worth of footnote material the material will be split resulting in footnotes out of order. This issue is now at least detected and generates an error but the algorithm used by the package is unable to gracefully handle it in an automated fashion (some alternatives for resolving the problem if it happens are given in the package documentation).

Improvement in `xspace` and font-switching The `xspace` package provides the command `\xspace` which attempts to be clever about inserting spaces automatically after user-defined control sequences. An important bug fix has been made to this command to correct its behaviour when used in conjunction with font-switching commands such as `\emph` and `\textbf`. Previously, writing

```
\newcommand\foo{foo\xspace}
... \emph{\foo}  bar baz
... \emph{\foo}, bar baz
```

would result in an extraneous space being inserted after ‘foo’ in both cases; this has now been corrected.

RTL in `multicol` The 1.7 release of `multicol` adds support for languages that are typeset right-to-left. For those languages the order of the columns on the page also needs to be reversed—something that wasn't possible in earlier releases.

<http://www.tug.org/TUGboat/tb27-2/tb87hagen-gyre.pdf>

The new feature is supported through the commands `\RLmulticolcolumns` (switching to right-to-left typesetting) and `\LRmulticolcolumns` (switching to left-to-right typesetting) the latter being the default.

Improve French babel interaction with varioref

Extracting and saving the page number turned out to be a source of subtle bugs. Initially it was done through an `\edef` with a bunch of `\expandafter` commands inside. This posed a problem if the page number itself contained code which needed protection (e.g., pr/4080) so this got changed in the last release to use `\protected@edef`. However, that in turn failed with Babel (bug report/4093) if the label contained active characters, e.g., a “:” in French. So now we use (after one failed attempt pr/4159) even more `\expandafter` commands and `\romannumeral` trickery to avoid any expansion other than what is absolutely required—making the code in that space absolutely unreadable.

```
\expandafter\def\expandafter#1\expandafter{%
\romannumeral
\expandafter\expandafter\expandafter
\z@
\expandafter \@cdr
\romannumeral
\expandafter\expandafter\expandafter
\z@
\csname r@#2\endcsname\@nil}%
```

Code like this nicely demonstrates the limitations in the programming layer of L^AT_EX 2_ε and the advantages that expl3 will offer on this level.

L^AT_EX News

Issue 21, May 2014

Scheduled L^AT_EX bug-fix release

This issue of L^AT_EX News marks the second bug-fix release of L^AT_EX 2_ε (standard L^AT_EX) since shifting to a new build system in 2009. Provided sufficient changes are made, we expect to make such releases yearly or every two years, in sync with T_EX Live.

Release notes

This release makes no changes to the core code in the L^AT_EX 2_ε format but there are a small number of documentation fixes (not listed here). In addition several packages in the **base** and **required** areas have been updated as detailed below.

This has been done in accordance with the philosophy of minimising problems in both forwards and backwards compatibility, so most of these changes should not be noticed by the regular L^AT_EX user.

References in the text below of the form “graphics/3873” are to bug reports listed at:
<http://latex-project.org/cgi-bin/ltxbugs2html>

fixltx2e updates

There are a number of bugs and faulty design decisions in L^AT_EX 2_ε that should have been corrected long ago in the kernel code. However, such corrections cannot be done as this would break backwards compatibility in the following sense. A large number of documents exist by now that have worked around the bug or have even made use of a particular misfeature. Thus changing the kernel code would break too many existing documents.

The corrections for these types of bug have therefore been collected together in a package that can be loaded only when needed; its name is **fixltx2e**. For this release we made the following changes to this package:

- Misspelled float placement specifiers such as `\begin{figure}[tv]` instead of `tb` are silently ignored by the kernel code. Now we test for such letters and issue an error message.
- L^AT_EX’s float handling algorithm can get out of sync if you mix single and double-column floats (as they are placed independently of each other). This was corrected in **fixltx2e** a few years ago but the fix was not perfect as one situation using `\enlargethispage` generated a low-level T_EX error. This behaviour of the package is now improved.

New fltrace package

For years the file `lATEX.output.dtx` contained some hidden code to trace the detailed behaviour of the float placement algorithm of L^AT_EX. Prompted by questions on StackExchange we now extract this code into a new **fltrace** package. To see the float algorithm in action (or to understand why it decides to place all your floats at the very end of the document) use

```
\usepackage{fltrace} \tracefloats
```

To stop tracing somewhere in the document use `\tracefloatsoff` and to see the current value of various float parameters use `\tracefloatvals`. As the package is identical to the kernel code with tracing added, it may or may not work if you load any other package that manipulates that part of the kernel code. In such a case your best bet is to load **fltrace** first.

inputenc package updates

The **inputenc** package allows different input encodings for L^AT_EX documents to be specified including the important **utf8** option used to specify the Unicode UTF-8 encoding. A common mistake in documents has been to also include this option when using the Unicode-based T_EX engines LuaT_EX and XeL^AT_EX producing strange errors as these engines natively deal with UTF-8 characters.

If a document stored in an 8bit encoding is processed by pdfT_EX, it needs the **inputenc** package to work correctly. However, if such a document is processed unchanged by LuaT_EX or XeL^AT_EX, then accented characters may silently get dropped from the output.

The package has been modified so that if used with LuaT_EX or XeL^AT_EX, then it just issues a warning if **utf8** or **ascii** is specified, and stops with an error for any other encoding requested.

One further improvement has been made to the encoding definition files (`.def`) used by **inputenc**: the catcode of `@` is now saved and restored when reading them instead of always using `\makeatother` inside the files (`latex/4192`).

The tools directory

In the past each of the sub-directories in the “required” section of the L^AT_EX distribution contained a single `.ins` file to generate the code files from the source files. We have now started to provide individual `.ins` files for each of those packages that are likely to require updates outside a major L^AT_EX release.

multicol updates

Version 1.8 of `multicol` implements some improvements/fixes and one extension. In the past the balancing algorithm enlarged the column height until it found a solution that satisfied all constraints. If there were insufficient break points then the final column height could have been much larger than expected and if that happened near the end of the page it resulted in the text overflowing into the bottom margin. This situation is now detected and in that case a normal page is cut and balancing is resumed on the next page. Some overflow is still allowed and controlled via the parameter `\maxbalancingoverflow`.

The use of `\enlargethispage` is now properly supported within the environment. Finally a new command `\docolaction` was added to allow the execution of code depending on the column in which the command is executed. See the documentation for details.

Bug fixes: the new version fixes both a color leak that could happen in certain situations and the problem that `multicols` could mess up the positioning of `\marginpars` that followed the environment.

tabularx updates

The restrictions on embedding `\tabularx` `\endtabularx` into the definition of a new environment have been relaxed slightly. See the package documentation for details.

showkeys updates

The `showkeys` package has been updated to fix problems if used at the start of list items, and to work if brace groups (`{` and `}`) are used in the optional argument of `\cite`. (`tools/4162`, `tools/4173`)

color updates

The `\nopagecolor` command suggested by Heiko Oberdiek, available for some years in the `pdftex` option, has been added to the core package as suggested in `graphics/3873`. Currently this is supported in the driver files for `dvips` and `pdftex`. Patches to support other drivers are welcome.

graphicx updates

The `graphicx` version of `\rotatebox` now allows `\par` (and blank lines) in values, to match the change made to the `graphics` version some years ago. See `graphics/4296`.

keyval updates

All parsing used in the `keyval` package has been changed to allow `\par` (and blank lines) in values. (A second change, to parsing of brace groups in a construct such as `key={{value}}`, was reverted in v1.15.) See `graphics/3446`.

Standard L^AT_EX (L^AT_EX 2_ε) and expl3

The substantial collection of innovative code in `expl3` implements a new programming language that has for a while now been used by some writers of L^AT_EX 2_ε packages. This code has recently also been made available for use on top of plain T_EX or ConT_EXt, largely to support generic packages that are supposed to work with different flavours of T_EX. These uses in no way affect authors of L^AT_EX documents and such L^AT_EX 2_ε packages will continue to work as advertised by their authors with standard L^AT_EX.

This code base will also become an important foundation for the kernel of L^AT_EX 3 and so the new programming language can be described as ‘The L^AT_EX 3 Programming Language’. However, if you see or hear that a package ‘uses L^AT_EX 3’ then it remains very unlikely (as yet) to mean that the package is part of some ‘new version of L^AT_EX’.

News about the development and use of `expl3` and about other developments in the L^AT_EX 3 code base is reported regularly in the L^AT_EX 3 News series (<http://latex-project.org/l3news/>), the most recent issue of which was published in March 2014.

L^AT_EX News

Issue 22, January 2015

New L^AT_EX 2_ε bug-fix policy

Introduction

For some years we have supplied bug fixes to the L^AT_EX 2_ε kernel via the `fixltx2e` package. This kept the kernel stable, but at the expense of meaning that most users did not benefit from bug fixes, and that some compromises which were made to save space in the machines of the time are still affecting most users today.

In this release we have started a new update policy. All the fixes previously available via `fixltx2e` are now enabled *by default* in the format, as are some further extensions for extended T_EX engines, ε -T_EX, X_ƎT_EX and LuaT_EX. Compatibility and stability are still important considerations, and while most users will not notice these improvements, or will want to benefit from them, a new `latexrelease` package is provided that will revert all the changes and re-instate the definitions from earlier releases. The package can also be used with older releases to effectively *update* the kernel to be equivalent to this 2015 release.

A new document, `latexchanges`, is distributed with the release that documents all the changes to documented commands since the 2014 L^AT_EX release, and will be updated in future releases if further changes have been made.

The `latexrelease` package

As noted above a new package is available to manage differences between L^AT_EX releases. If you wish to revert all changes back to the definitions as they were in previous releases you may start your document requesting the L^AT_EX release from May 2014:

```
\RequirePackage[2014/05/01]{latexrelease}
\documentclass{article}
```

Conversely if you start a large project now and want to protect yourself against possible future changes, you may start your document

```
\RequirePackage[2015/01/01]{latexrelease}
\documentclass{article}
```

Then the version of `latexrelease` distributed with any future L^AT_EX release will revert any changes made in that format, and revert to the definitions as they were at the beginning of 2015.

If you wish to share a document using the latest features with a user restricted to using an older format, you may use the form above and make the `latexrelease`

package available on the older installation. The package will then update the format definitions as needed to enable the older format to work as if dated on the date specified in the package option.

The `\IncludeInRelease` command

The mechanism used in the `latexrelease` package is available for use in package code. If in your `zzz` package you have

```
\RequirePackage{latexrelease}
\IncludeInRelease{2015/06/01}
  {\zzz}{\zzz definition}
  \def\zzz.....new code
\EndIncludeInRelease
\IncludeInRelease{0000/00/00}
  {\zzz}{\zzz definition}
  \def\zzz....original
\EndIncludeInRelease
```

then in a document using a format dated 2015/06/01 or later, the “new code” will be used, and for documents being processed with an older format, the “original” code will be used. Note the format date here may be the original format date as shown at the start of every L^AT_EX run, or a format date specified as a package option to the `latexrelease` package.

So if the document has

```
\RequirePackage[2014/05/01]{latexrelease}
\documentclass{article}
\usepackage{zzz}
```

then it will use the *original* definition of `\zzz` even if processed with the current format, as the format acts as if dated 2014/05/01.

Limitations of the approach

The new concept provides full backward and forward compatibility for the L^AT_EX format, i.e., with the help of a current `latexrelease` package the kernel can emulate all released formats (starting with 2014/06/01¹).

However, this is not necessarily true for all packages. Only if a package makes use of the `\IncludeInRelease` functionality will it adjust to the requested L^AT_EX release date. Initially this will only be true for a few selected packages and in general it may not even be

¹Patching an older format most likely works too, given that the changes in the past have been minimal, though this isn’t guaranteed and hasn’t been tested.

advisable for packages that have their own well-established release cycles and methods.

Thus, to regenerate a document with 100 % compatible behavior it will still be necessary to archive it together with all its inputs, for example, by archiving the base distribution trees (and any modifications made). However, the fact that a document requests a specific L^AT_EX release date should help identifying what release tree to use to achieve perfect accuracy.

Updates to the kernel

Updates incorporated from fixltx2e

The detailed list of changes incorporated from fixltx2e is available in the new latexchanges document that is distributed with this release. The main changes are that 2-column floats are kept in sequence with one column floats, corrections are made to the \mark system to ensure correct page headings in 2-column documents, several additional commands are made robust.

ε -T_EX register allocation

L^AT_EX has traditionally used allocation routines inherited from plain T_EX that allocated registers in the range 0–255. Almost all distributions have for some years used ε -T_EX based formats (or X_YT_EX or LuaT_EX) which have 2¹⁵ registers of each type (2¹⁶ in the case of LuaT_EX). The etex package has been available to provided an allocation mechanism for these extended registers but now the format will by default allocate in a range suitable for the engine being used. The new allocation mechanism is different than the etex package mechanism, and supports LuaT_EX’s full range and an allocation mechanism for L^AT_EX floats as described below.

On ε -T_EX based engines, an additional command, \newmarks is available (as with the etex package) that allocates extended ε -T_EX marks, and similarly if X_YT_EX is detected a new command \newXeTeXintercharclass is available, this is similar to the command previously defined in the xelatex.ini file used to build the xelatex format.

Additional L^AT_EX float storage

L^AT_EX’s float placement algorithm needs to store floats (figures and tables) until it finds a suitable page to output them. It allocates 18 registers for this storage, but this can often be insufficient. The contributed morefloats package has been available to extend this list; however, it also only allocates from the standard range 0–255 so cannot take advantage of the extended registers. The new allocation mechanism in this release incorporates a new command \extrafloats. If you get the error: Too many unprocessed floats. then you can add (say) \extrafloats{500} to the document

preamble to make many more boxes available to hold floats.

Built-in support for Unicode engines

The kernel sources now detect the engine being used and adjust definitions accordingly, this reduces the need for the “.ini” files used to make the formats to patch definitions defined in latex.ltx.

As noted above the format now includes extended allocation routines.

The distribution includes a file unicode-letters.def derived from the Unicode Consortium’s Unicode Character Data files that details the upper and lower case transformation data for the full Unicode range. This is used to set the lccode and uccode values if a Unicode engine is being used, rather than the values derived from the T1 font encoding which are used with 8-bit engines.

Finally \typein is modified if LuaT_EX is detected such that it works with this engine.

l3build

This release has been tested and built using a new build system implemented in Lua, intended to be run on the texlua interpreter distributed with modern T_EX distributions. It is already separately available from CTAN. This replaces earlier build systems (based at various times on make, cons, and Windows bat files). It allows the sources to be tested and packaged on a range of platforms (within the team, OS X, Windows, Linux and Cygwin platforms are used). It also allows the format to be tested on X_YT_EX and LuaT_EX as well as the standard pdfT_EX/ ε -T_EX engines.

Hyperlinked documentation and TDS zip files

As well as updating the build system, the team have looked again at exactly what gets released to CTAN. Taking inspiration from Heiko Oberdiek’s latex-tds bundle, the PDF documentation provided now includes hyperlinks where appropriate. This has been done without modifying the sources such that users without hyperref available can still typeset the documentation using only the core distribution. At the same time, the release now includes ready-to-install TDS-style zip files. This will be of principal interest to T_EX system maintainers, but end users with older machines who wish to manually update L^AT_EX will also benefit.

L^AT_EX News

Issue 23, October 2015

Contents

Enhanced support for LuaT_EX

As noted in L^AT_EX News 22, the 2015/01/01 release of L^AT_EX introduced built-in support for extended T_EX systems.

The range of allocated register numbers (for example, for count registers) is now set according to the underlying engine capabilities to 256, 32768 or 65536. Additional allocators were also added for the facilities added by ε -T_EX (`\newmark`) and X_YT_EX (`\newXeTeXintercharclass`). At that time, however, the work to incorporate additional allocators for LuaT_EX was not ready for distribution.

The main feature of this release is that by default it includes allocators for LuaT_EX-provided features, such as Lua functions, bytecode registers, catcode tables and Lua callbacks. Previously these features have been provided by the contributed `luatex` (Heiko Oberdiek) and `luatexbase` (Élie Roux, Manuel Pégourié-Gonnard and Philipp Gesang) packages. However, just as noted with the `etex` package in the previous release, it is better if allocation is handled by the format to avoid problems with conflicts between different allocation schemes, or definitions made before a package-defined allocation scheme is enabled.

The facilities incorporated into the format with this release, and described below, are closely modelled on the `luatexbase` package and we thank the authors, and especially Élie Roux, for help in arranging this transition.

The implementation of these LuaT_EX features has been redesigned to match the allocation system introduced in the 2015/01/01 L^AT_EX release, and there are some other differences from the previous `luatexbase` package. However, as noted below, `luatexbase` is being updated in line with this L^AT_EX release to provide the previous interface as a wrapper around the new implementation, so we expect the majority of documents using `luatexbase` to work without change.

Names of LuaT_EX primitive commands

The 2015/01/01 L^AT_EX release for the first time initialised LuaT_EX in `latex.ltx` if LuaT_EX is being used. Following the convention used in the contributed `lualatex.ini` file used to set up the format for earlier releases, most LuaT_EX-specific primitives were defined with names prefixed by `luatex`. This was designed to

minimize name clashes but had the disadvantage that names did not match the LuaT_EX manual, or the names used in other formats, and produced some awkward command names such as `\luatexluafunction`. From this release the names are enabled without the `luatex` prefix.

In practice this change should not affect many documents; relatively few packages access the primitive commands, and many of those are already set up to work with prefixed or unprefixed names, so that they work with multiple formats.

For package writers, if you want to ensure that your code works with this and earlier releases, use unprefixed names in the package and ensure that they are defined by using code such as:

```
\directlua{tex.enableprimitives("",
    tex.extraprimitives(
        "omega", "aleph", "luatex"))}
```

Conversely if your document uses a package relying on prefixed names then you can add:

```
\directlua{tex.enableprimitives("luatex",
    tex.extraprimitives(
        "omega", "aleph", "luatex"))}
```

to your document.

Note the compatibility layer offered by the `luatexbase` package described below makes several commands available under both names.

As always, this change can be reverted using:
`\RequirePackage[2015/01/01]{latexrelease}`
at the start of the document.

T_EX commands for allocation in LuaT_EX

For detailed descriptions of the new allocation commands see the documented sources in `lualatex.dtx` or chapter N of `source2e`; however, the following new allocation commands are defined by default in LuaT_EX: `\newattribute`, `\newcatcodetable`, `\newluafunction` and `\newwhatsit`. In addition, the commands `\setattribute` and `\unsetattribute` are defined to set and unset Lua attributes (integer values similar to counters, but attached to nodes). Finally several catcode tables are predefined: `\catcodetable@initex`, `\catcodetable@string`, `\catcodetable@latex`, `\catcodetable@atletter`.

Predefined Lua functions

If used with LuaTeX, L^AT_EX will initialise a Lua table, `luatexbase`, with functions supporting allocation and also the registering of Lua callback functions.

Support for older releases and plain T_EX

The LuaTeX allocation functionality made available in this release is also available in plain T_EX and older L^AT_EX releases in the files `luatex.tex` and `luatex.lua` which may be used simply by including the T_EX file: `\input{luatex}`. An alternative for old L^AT_EX releases is to use:

```
\RequirePackage[2015/10/01]{latexrelease}
```

which will update the kernel to the current release, including LuaTeX support.

Additional LuaTeX support packages

In addition to the base L^AT_EX release two packages have been contributed to the `contrib` area on CTAN. The `ctablestack` package offers some commands to help package writers control the LuaTeX `catcodetable` functionality, and the `luatexbase` package replaces the previously available package of the same name, providing a compatible interface but implemented over the `luatex` code.

More Floats and Inserts

If ϵ -T_EX is available, the number of registers allocated in the format to hold floats such as figures is increased from 18 to 52.

The extended allocation system introduced in 2015/01/01 means that in most cases it is no longer necessary to load the `etex` package. Many classes and packages that previously loaded this package no longer do so. Unfortunately in some circumstances where a package or class previously used the `etex` `\reserveinserts` command, it is possible for a document that previously worked to generate an error “no room for a new insert”. In practice this error can always be avoided by declaring inserts earlier, before the registers below 256 are all allocated. However, it is better not to require packages to be re-ordered and in some cases the re-ordering is complicated due to delayed allocations in `\AtBeginDocument`.

In this release, a new implementation of `\newinsert` is used which allocates inserts from the previously allocated float lists once the classical register allocation has run out. This allows an extra 52 (or in LuaTeX, 64 thousand) insert allocations which is more than enough for practical documents (by default, L^AT_EX only uses two insert allocations).

Updated Unicode data

The file `unicode-letters.def` recording catcodes, upper and lower case mappings and other properties for

Unicode characters has been regenerated using the data files from Unicode 8.0.0.

Support for Comma Accent

The command `\textcommabelow` has been added to the format. This is mainly used for the Romanian letters ȘșȚț. This was requested in latex/4414 in the L^AT_EX bug tracker.

Extended inputenc

The `utf8` option for `inputenc` has been extended to support the letters s and t with comma accent, U+0218–U+021b. Similarly circumflex w and y U+0174–U+0177 are defined. Also U+00a0 and U+00ad are declared by default, and defined to be `\nobreakspace` and `\-` respectively.

The error message given on undefined UTF-8 input characters now displays the Unicode number in U+*hex* format in addition to showing the character.

Pre-release Releases

The patch level mechanism has been used previously to identify L^AT_EX releases that have small patches applied to the main release, without changing the main format date.

The mechanism has now been extended to allow identification of pre-release versions of the software (which may or may not be released via CTAN) but can be identified with a banner such as

```
LaTeX2e <2015/10/01> pre-release-1
```

Internally this is identified as a patch release with a negative patch level.

Updates in tools

The `multicol` package has been updated to fix the interaction with “here” floats that land on the same page as the start or end of a `multicols` environment.

L^AT_EX News

Issue 24, February 2016

Contents

LuaT_EX support	30
Unicode data	30
More support for east European accents	31
Changes in Graphics	31
Changes in Tools	31
Improving support for Unicode engines	31

LuaT_EX support

This release refines the LuaT_EX support introduced in the 2015/10/01 release. A number of patches have been added to improve the behavior of `lATEX` (thanks largely to code review by Philipp Gesang). The kernel code has been adjusted to allow for changes in LuaT_EX v0.85–v0.88. Most notably, newer LuaT_EX releases allow more than 16 write streams and these are now enabled for use by `\newwrite`, but also the experimental `newtoken` Lua library has been renamed back to `token` which required small adjustments in the LuaT_EX setup.

The biggest change in LuaT_EX v0.85–v0.87 compared to previous versions is that all the primitives (originally defined in `pdfTEX`) dealing with the PDF “back end” are no longer defined, being replaced by a much smaller set of new primitives. This does not directly affect the core L^AT_EX files in this release but has required major changes to the `.ini` files used by T_EX Live and similar distributions to set up the format files. These changes in the LuaT_EX engine will affect any packages using these back end commands (packages such as `graphics`, `color`, `hyperref`, etc.). Until all contributed packages are updated to the new syntax users may need to add aliases for the old `pdfTEX` commands. A new `luapdftexalias` package has been contributed to CTAN (not part of the core L^AT_EX release) that may be used for this purpose.

See also the sections below for related changes in the tools and `graphics` bundles.

Unicode data

As noted in L^AT_EX News 22, the 2015/01/01 release of L^AT_EX introduced built-in support for extended T_EX systems. In particular, the kernel now loads appropriate

data from the Unicode Consortium to set `\lccode`, `\uccode`, `\catcode` and `\sfcode` values in an automated fashion for the entire Unicode range.

The initial approach taken by the team was to incorporate the existing model used by (plain) X_YT_EX and to pre-process the “raw” Unicode data into a ready-to-use form as `unicode-letters.def`. However, the relationship between the Unicode Consortium files and T_EX data structures is non-trivial and still being explored. As such, it is preferable to directly parse the original (`.txt`) files at point of use. The team has therefore “spun-out” both the data and the loading to a new generic package, `unicode-data`. This package makes the original Unicode Consortium data files available in the `texmf` tree (in `tex/generic/unicode-data`) and provides generic loaders suitable for reading this data into the plain, L^AT_EX 2_ε, and other, formats.

At present, the following data files are included in this new package:

- `CaseFolding.txt`
- `EastAsianWidth.txt`
- `LineBreak.txt`
- `MathClass.txt`
- `SpecialCasing.txt`
- `UnicodeData.txt`

These files are used either by L^AT_EX 2_ε or by `expl3` (i.e. they represent the set currently required by the team). The Unicode Consortium provides various other data files and we would be happy to add these to the generic package, as it is intended to provide a single place to collect this material in the `texmf` tree. Such requests can be mailed to the team as usual or logged at the package home page: <https://github.com/latex3/unicode-data>.

The new approach extends use of Unicode data in setting T_EX information in two ways. First, the `\sfcode` of all end-of-quotation/closing punctuation is now set to 0 (transparent to T_EX). Second, `\Umathcode` values are now set using `MathClass.txt` rather than setting up only letters (which was done using an arbitrary plane 0/plane 1 separation). There are also minor refinements to the existing code setting, particularly splitting the concepts of case and letter/non-letter category codes.

For X_YT_EX, users should note that `\xtxHanGlue` and `\xtxHanSpace` are *no longer defined*, that no

assignments are made to `\XeTeXinterchartoks` and that no `\XeTeXintercharclass` data is loaded into the format. The values which were previously inherited from the plain \TeX setup files are *not* suitable for properly typesetting East Asian text. There are third-party packages addressing this area well, notably those in the C \TeX bundle. Third-party packages may need adjustment to load the data themselves; see the `unicode-data` package for one possible loader.

More support for east European accents

As noted in L \TeX News 23, comma accent support was added for `s` and `t` in the 2015/10/01 release. In this release a matching `\textcommaabove` accent has been added for U+0123 (`\c{g}`, `g`) which is the lower case of U+0122 (`\c{G}`, `G`). In the OT1 and T1 encodings the combinations are declared as composites with the `\c` command, which matches the Unicode names “latin (capital|small) letter g with cedilla” and also allows `\MakeUppercase{\c{g}}` to produce `\c{G}`, as required. In T1 encoding, the composite of `\c` with `k`, `l`, `n` and `r` are also declared to use the comma below accent rather than cedilla to match the conventional use of these letters.

The UTF-8 `inputenc` option `utf8` has been extended to support all latin combinations that can be reasonably constructed with a (single) accent command and a base character for the T1 encoding so `g`, `u` and similar characters may be directly input using UTF-8 encoding.

Changes in Graphics

The changes in Lua \TeX v0.87 mean that the `color` and `graphics` packages no longer share the `pdftex.def` file between Lua \TeX and pdf \TeX . A separate file `luatex.def` (distributed separately) has been produced, and distributions are encouraged to modify `graphics.cfg` and `color.cfg` configuration files to default to the `luatex` option if Lua \TeX v0.87 or later is being used. The team has contributed suitable `.cfg` files to CTAN to be used as models.

Normally it is best to let the local `graphics.cfg` automatically supply the right option depending on the \TeX engine being used; however the `color` and `graphics` (and so `graphicx`) packages have been extended to have an explicit `luatex` option comparable to the existing `pdftex` and `xetex` options.

The `trig` package has been updated so that pre-computed values such as `\sin(90)` now expand to digits (1 rather than the internal token `\@one` in this case). This allows them to be used directly in PDF literal strings.

Changes in Tools

Lua \TeX from version v0.87 no longer supports the

`\write18` syntax to access system commands. A new package `shellesc` has been added to `tools` that defines a new command `\ShellEscape` that may be used in all \TeX variants to provide a consistent access to system commands. The package also defines `\write18` in Lua \TeX so that it continues to access system commands as before; see the package documentation for details.

Improving support for Unicode engines

Stability concerns are always paramount when considering any change to the L \TeX 2 ϵ kernel. At the same time, it is important that the format remains usable and gives reliable results for users. For the Unicode \TeX engines X \TeX and Lua \TeX there are important differences in behavior from classical (8-bit) \TeX engines which mean that identical default behaviors are not appropriate. Over the past 18 months the team has addressed the most pressing of these considerations (as detailed above and in L \TeX News 22 and 23), primarily by integrating existing patches into the kernel. There are, though, important areas which still need consideration, and which *may* result in refinements to kernel support in this area in future releases.

The default font setup in L \TeX 2 ϵ at present is to use the OT1 encoding. This assumes that hyphenation patterns have been read using appropriate codes: the T1 encoding is assumed. The commonly-used hyphenation patterns today, `hyph-utf8`, are set up in this way for 8-bit engines (pdf \TeX) but for Unicode engines use Unicode code points. This means that hyphenation will be incorrect with Unicode engines unless a Unicode font is loaded. This requires a concept of a Unicode font encoding, which is currently provided by the `fontspec` package in two versions, EU1 and EU2. The team is working to fully understand what is meant by a “Unicode font encoding”, as unlike a classical \TeX encoding it is essentially impossible to know what glyphs will be provided (though each slot is always defined with the same meaning). There is also an overlap between this area and ideas of language and writing system, most obviously in documents featuring mixed scripts (for example Latin and Cyrillic).

As well as these font considerations, the team is also exploring to what extent it is possible to allow existing (8-bit) documents to compile directly with Unicode engines without requiring changes in the sources. Whether this is truly possible remains an open question.

It is important to stress that changes will only be made in this area where they do *not* affect documents processed with ϵ - \TeX /pdf \TeX (i.e. documents which are written for “classical” 8-bit \TeX engines). Changes will also be made only where they clearly address deficiencies in the current setup for Unicode engines

(i.e. where current behaviors are wrong).