

# A $\LaTeX$ Tour, part 1: the base distribution

David Carlisle

## 1 Introduction

In this article<sup>1</sup> I hope to give a ‘guided tour’ around the files that make up the basic  $\LaTeX$  distribution. Subsequent articles in this mini-series will cover other packages by the  $\LaTeX$  development team, and also some of the main contributed packages.

The primary source for  $\LaTeX$  is the ‘CTAN’<sup>2</sup> network of archives, so if I refer to path names of files this relates to the CTAN file structure. Note however that if you obtained  $\LaTeX$  as part of a ‘pre-packaged’  $\TeX$  distribution, then these files may have been moved (typically documentation files may be separated from  $\TeX$  source files). Hopefully this will not cause any confusion.

## 2 The Components of $\LaTeX$

The  $\LaTeX$  distribution at the CTAN archives is organised into the following directories.

`base` Contains the core  $\LaTeX$  files. You need at least these files for a minimal  $\LaTeX$  installation.

`unpacked` Includes *all* the files in `base` together with the result of ‘unpacking’ the source files. (More about this later.) Thus when obtaining  $\LaTeX$  you should get either `base` or `unpacked`, but not both. Getting the former saves on time transferring the files, but getting the latter saves time that would be taken unpacking the source files, so which is preferable depends on the relative speed (and cost) of your machine and your connection to the archives.

`packages` Consists of seven independent  $\LaTeX$  ‘extensions’ that are written and supported by the  $\LaTeX$  developers (or the American Mathematical Society).

`amsfonts`, `amslatex`, `babel`, `graphics`, `mfnfss`, `psnfss` and `tools`

These packages will be described in more detail later in the series.

`fonts` The metafont sources and  $\TeX$  font metric files of a few fonts that  $\LaTeX$  requires that are not part of the original plain  $\TeX$  distribution.

`doc` This directory is not part of the main  $\LaTeX$  distribution, it is generated by the CTAN archives. As a convenience for those people that have not yet installed  $\LaTeX$ , some of the main introductory doc-

umentation files which are available as  $\LaTeX$  files in the base distribution are made available in this directory as `dvi` and `POSTSCRIPT` files.

`contrib` This directory contains an ever growing number of contributed  $\LaTeX$  packages, and other extensions, that have been contributed by  $\LaTeX$  users. They are not part of the ‘official’  $\LaTeX$  distribution, but many of them form a vital part of any ‘working’  $\LaTeX$  installation. The packages are divided into two subdirectories ‘supported’ and ‘other’, however at the current time one should ignore this distinction when looking for packages to fetch from the archives. Contrary to expectations some of the best supported packages are distributed (at their author’s request) from `contrib/other`.

Unfortunately (for mainly historical reasons) people in search of contributed  $\LaTeX$  packages also need to look in more distant CTAN directories. Firstly, the `macros/latex209/contrib` area on CTAN contains packages that were written for the previous version of  $\LaTeX$  that has been obsolete since June 1994. Any packages that are still distributed from this  $\LaTeX$  2.09 tree are likely to be less than well supported, but you can still find some useful files there. Secondly, there are some packages that work with multiple formats, not just  $\LaTeX$ , and these are to be found under `macros/generic` or in their own top-level directory, such as `macros/musictex`.

## 3 Documentation in the Base Distribution

The documentation that comes with  $\LaTeX$  is of two forms: plain (ascii) text files with extension `.txt`, or  $\LaTeX$  documents distributed as  $\LaTeX$  source with extension `.tex`. Generally speaking the text files are mainly of interest to people installing  $\LaTeX$ , who may need information before they have a working system. Information of more general interest to  $\LaTeX$  users is normally distributed as a  $\LaTeX$  document.

### 3.1 The ASCII text files

#### 3.1.1 Installation instructions

`00readme` Provides a general introduction to the system, and should be the first file to look at when installing  $\LaTeX$  for the first time.

<sup>1</sup>This is a slightly edited version of an article that first appeared in the UKTUG journal *Baskerville* 6.1.

<sup>2</sup>The Comprehensive  $\TeX$  Archive Network:

`ftp.tex.ac.uk` (UK), `ftp.dante.de` (Germany), in the USA the mirror `ftp.cdrom.com` of the UK archive should be used in preference to the original CTAN host of `ftp.shsu.edu` which is unfortunately unreliable at the present time.

- `install` Provides ‘generic’ installation instructions, but for many  $\TeX$  versions more specific instructions that have been contributed by the authors or users of those systems; thus `emtex` gives instructions for the popular `em $\TeX$`  implementation, `web2ctex` gives specific instructions on installing under UNIX, etc.
- `tex2` If you have a  $\TeX$  that pre-dates version 3.0 (which was released in 1989) by far the best thing to do is to update your  $\TeX$ , but if that is really not possible this file details how  $\LaTeX$  may be built under  $\TeX$ 2.
- `autoload` Describes the installation of an ‘autoloading’ version of  $\LaTeX$ . This produces a much smaller format by saving less common commands in external files rather than in memory. These files are automatically ‘autoloaded’ as required. This version of  $\LaTeX$  is particularly recommended if you are using a small installation (for instance a ‘small’ `em $\TeX$`  on a sub-386 PC).

### 3.1.2 Other text files

- `legal` Contains the copyright notices and distribution conditions for  $\LaTeX$ .
- `bugs` Contains instructions on how to compile a bug report (see below).
- `patches` Describes the  $\LaTeX$  patch mechanism that is used for distributing small updates between the ‘full’ releases. This file also contains a list of all the files that have changed since the last full release.
- `changes` A Change Log of all the changes made to the  $\LaTeX$  files. This is mainly intended for internal use by the  $\LaTeX$  developers, but some people like to read it.

## 3.2 The $\LaTeX$ ‘guides’

These documents are distributed as  $\LaTeX$  source (i.e., `.tex` files) although as noted in the above introduction, the CTAN archives distribute most of them in ready-formatted versions in the directory `latex/doc` so you can read these before installing  $\LaTeX$  if you wish. Unlike the ASCII text files described above, most of these documents are primarily intended for *users* of the system rather than system managers and software installers.

- `usrguide`  *$\LaTeX$ 2 $\epsilon$  for Authors*. This document describes all the main new features of the 2 $\epsilon$  release of  $\LaTeX$ . It was written originally with the user of the old  $\LaTeX$  2.09 in mind, but newcomers to  $\LaTeX$  who have never used the old version should still gain something by reading this document. It does not however cover the majority of  $\LaTeX$  commands that were not changed, and so it is not a substitute for a full  $\LaTeX$  manual.
- `clsguide`  *$\LaTeX$ 2 $\epsilon$  for class and package writers*. A companion to `usrguide`, gives details of the  $\LaTeX$  commands for structuring class files and extension packages.

`fntguide`  *$\LaTeX$ 2 $\epsilon$  font selection*. For font addicts only, but if you want to know the detailed specification of the ‘New Font Selection Scheme’ commands, here is the place to look.

`cfgguide` *Configuration options for  $\LaTeX$ 2 $\epsilon$* . Discusses what you can (and can not) do to configure a  $\LaTeX$  installation to the requirements of your local site.

`ltx3info` *The  $\LaTeX$ 3 Project*. A brief summary of the aims of the  $\LaTeX$ 3 project, the group of volunteers that has taken on the maintenance and development of  $\LaTeX$ .

`modguide` *Modifying  $\LaTeX$* . This document discusses some of the rationale behind the  $\LaTeX$  distribution conditions as expressed in `legal.txt` and `cfgguide.tex`. Unless you are making a distribution of a modified version of  $\LaTeX$ , or are particularly interested in software copyright issues, you probably do not want to read this.

## 3.3 $\LaTeX$ News

As well as these larger documents there are a series of one-page ‘newsletters’. A new one is produced with each full release of  $\LaTeX$ . These detail any changes that have occurred in  $\LaTeX$  or the main extension packages over the six months since the previous release. ( $\LaTeX$  releases occur at regular intervals, in June and December of each year.) Currently the four files `ltnews01–ltnews04` are distributed; these correspond to the four releases of  $\LaTeX$  since June 1994.

## 3.4 Example Documents

There are two (very) small example documents, as described in the  $\LaTeX$  book by Leslie Lamport.

- `small2e` A very small (1 page)  $\LaTeX$  document.
- `sample2e` A slightly larger document.

## 3.5 Documented sources

The source for the  $\LaTeX$  format, and for all the packages and classes in the core distribution is distributed as ‘`dtx`’ files. These are  $\LaTeX$  documents which may be processed in the usual way to produce typeset documentation. For example a command such as `latex ltpictur.dtx` would produce documented source of the picture mode commands. The files with names of the form ‘`lt...dtx`’ make up the source of the  $\LaTeX$  format. If you want to produce a combined document incorporating all these files, you may process `source2e.tex`. This document will produce a typeset version of the  $\LaTeX$  sources, together with change log and index. It is well over 500 pages long, and so may take a long time to produce. It may produce an index that is too large to be handled by the ‘`makeindex`’ program on smaller machines.

## 3.6 Errata

The principal documentation for  $\LaTeX$  is the two books  *$\LaTeX$ : A Document Preparation System*, and *The  $\LaTeX$  Companion*. Errata for these (and the German edi-

tion of *The Companion*) are available as `manual.err`, `compan.err` and `begleit.err`.

## 4 The $\LaTeX$ Bug Report Database

As described in the file `bugs.txt` mentioned above, the  $\LaTeX$ 3 project maintain a database of bug reports for  $\LaTeX$ .

If, after checking with colleagues, reading the manual, etc., you decide that some behaviour of  $\LaTeX$  is incorrect then you may send a message to the  $\LaTeX$  bug database. Before doing this you should check that your  $\LaTeX$  is not more than one year old (the bug may have been fixed in a recent release). If you have access to the World Wide Web, you may access the database and see if the problem is already reported by using the search page accessible from: <http://www.tex.ac.uk/CTAN/latex/bugs.html>.

If you decide to send a report, two files are available to help compose a message in the correct format:

`latexbug.tex`  $\LaTeX$  this file and you will be prompted for information such as your name, and the name of a test file that shows the problem. A mail message will be written to the file `latexbug.msg` which should be sent to `latexbugs@uni-mainz.de`. (You should *always* use `latexbug.tex` to generate messages to be sent to this bug address. It is an interface to a database (the GNU GNATS problem tracking system) and can not handle messages that are not in the special format written by `latexbug.tex`.

`latexbug.el` For users of the GNU Emacs text editor, a more convenient interface is provided by this file. It runs `latexbug.tex` automatically, and provides online help for filling in the various fields, and finally automatically mails the message to the correct address.

## 5 Docstrip files

As mentioned above,  $\LaTeX$  is distributed as documented sources. The files that are actually used by  $\TeX$  are extracted from these files by running `docstrip.tex`. The  $\LaTeX$  distribution contains many files with extension `.ins` that control how `docstrip` extracts each file. Most of these are never used individually, as they would just ‘unpack’ one small part of the distribution. The file `unpack.ins` is a ‘master’ installation script that calls the smaller install files in turn and so unpacks the whole distribution. Normally running  $\TeX$  on this file is the first step in installing  $\LaTeX$ . This step may be omitted however if the unpacked directory is obtained from CTAN rather than base. `unpack` is *exactly* the result of obtaining base and running  $\TeX$  on `unpack.ins`. If you have a slow machine you may prefer this route as it saves unpacking time, but conversely it requires downloading more files, so if you are transferring the files via a slow connection such

as a modem then you may prefer to get the smaller ‘base’ distribution.

There are three install files that are *not* included into `unpack.ins` so you may have need to run these if you need the following features.

`autoload` Processing `autoload.ins` will generate the source file for the ‘autoload’ version of  $\LaTeX$ , `latexa.ltx`, as described in `autoload.txt`. This should be processed with `iniTeX` to create a format file to be used in place of the standard `latex.fmt`. As well as the modified format, various packages are created containing the code that has been taken out of the format. Normally these do not need to be invoked explicitly as they are loaded on demand when they are needed. Currently the following package files are produced.

`autopict` Source for `picture` mode.

`autotabg` Source for `tabbing` environment.

`autoerr` The texts of most  $\LaTeX$  error commands.

`autofss1` Less used font selection commands.

`autoout1` Code related to `\enlargethispage`.

The `autoload` format is still quite experimental, and so the range of such ‘autoloading’ packages may change with future releases.

`cmextra` Processing `cmextra.ins` installs the ‘fd’ files for the ‘concrete’ variants of the Computer Modern fonts, and also the AMS Cyrillic fonts.

`olddc` If using the Computer Modern fonts in the 8-bit ‘T1’ encoding,  $\LaTeX$  defaults to using the ‘dc fonts’. During 1995 these fonts were updated and the names of the fonts *changed*. Thus the 10pt roman font corresponding to `cmr10` is now `dcr1000` rather than `dcr10`. The install file `unpack.ins` includes `newdc.ins` so by default  $\LaTeX$  will use the new 1995 names (dc fonts release 1.2 or later) when using T1 font encoding. If you still have the old dc fonts, then you must process `olddc.ins` to produce suitable fd files referring to the old names.

## 6 The Standard $\LaTeX$ Classes

The general appearance of a  $\LaTeX$  document, and the specification of the commands available is specified in a *document class*. This may be further modified by loading *packages*, as described in `usrguide`. In this section I give a brief overview of the available classes in the base distribution. They all have extension `.cls` (after being unpacked from the `.dtx` source file during the installation process).

`article` ‘Article Class’. In some sense the canonical reference class against which all others are judged. This class (which is generated from the same `classes.dtx` source as `report` and `book` described below) is a mixed blessing. On one hand

it provides quite a rich collection of commands for marking up documents that means that it serves well as the basic ‘generic’ class to be used when no more suitable specific class is available. On the other hand the visual appearance of documents produced with this class is very distinctive. Many people who say they “don’t like  $\LaTeX$ ” and so use some other format such as plain, in fact are misled into believing that  $\LaTeX$  is this class. In fact by loading `article` and then making small adjustments one can produce very different visual designs. The class files for *Baskerville* and TUGBOAT are examples of such non-standard classes based on `article`.

However for many purposes, portability is more important than original typographical design, and in these cases the `article` class has the big advantage of being installed at all  $\LaTeX$  sites.

`report` ‘Report Class’. Very similar to `article` (and produced from the same source). The main differences are that this class has a higher level of sectioning command (`\chapter`) than is available in `article`, and the front matter is typeset differently.

`book` The `book` class is again very similar to `report` with the addition of a few extra features for controlling the front matter and back matter. It is unlikely that you would want to use this class ‘as is’ for a book, as you would almost certainly want to spend some effort (and perhaps money!) on an original design. However it can be used as a basis or example of the implementation of a  $\LaTeX$  class for book production.

`letter` This provides commands for producing one or more letters. Many sites use this as a basis for producing site-specific letter class files, for instance with a modified heading that inserts a departmental logo and address.<sup>3</sup>

`proc` Proceedings class. This is a variant of `article` class (and inputs the `article.cls` file when used). It defaults to two column mode and makes one or two other small adjustments. It may be used as a model for how make a class that builds on another.

`slides` The `slides` class. This class essentially provides the functionality that was formally built into  $\text{SL}\TeX$ . It provides a mechanism for producing pages suitable for projecting on an overhead projector. It is described in the  $\LaTeX$  book, and some people like it; however if you are making a lot of such presentations you may prefer to look at the contributed classes `seminar` (T. v. Zandt) or `foiltex` (J. Hafner). These provide alternatives to the standard class that many people find more useful.

As well as these ‘Standard Classes’ the base distribution contains a few other special purpose classes.

`minimal` This is the minimal  $\LaTeX$  class. It just sets up a text area, and a font in a single size. None of

the normal sectioning or font size commands are available. This class is not intended to be used in documents, but it is often useful when testing macros as it loads very quickly.

`ltxguide` A special purpose class for the ‘ $\LaTeX$  guides’ mentioned earlier.

`ltnews` The class file used for the ‘ $\LaTeX$  News’ news sheets.

`ltxdoc` This class is used in all the `dtx` documentation files. It is based on the `article` class and the `doc` package, but with additional commands for documenting the  $\LaTeX$  sources. It was not conceived as a class for general use, but some people find it convenient to use it when documenting their own package files.

## 7 Standard Packages

### 7.1 Encoding Packages

One of the main features of the 2e release of  $\LaTeX$  is that it attempts to remove all ‘hard wired’ assumptions about the encodings being used, both for input and also in the fonts used for typesetting.

It maintains a strict distinction between the *Input Encoding* and the *Output Encoding*. The input encoding relates to the text that you type; this may be a standard encoding such as ASCII (the traditional 7-bit encoding) or ISO-latin-1, or a platform-specific encoding such as ‘Windows ANSI’ as used on MicroSoft Windows 3.x machines. The output encoding for text fonts is usually either OT1 (the encoding devised by Knuth and implemented in the original Computer Modern  $\TeX$  fonts.) or T1 (the new  $\TeX$  encoding also known as ‘Cork’ after the meeting where it was agreed).

$\LaTeX$  maintains this separation by *always* translating input to an *Internal Encoding*. This is essentially traditional  $\TeX$  7-bit input. This internal encoding is then translated to the encoding used in the font without reference to the original input mechanism used. Thus if you specify an input encoding that includes the character  $\acute{e}$  you may type that directly at the keyboard, and see it as a single character; however internally  $\LaTeX$  will treat this as `\' {e}`. If you are using 7-bit OT1 encoded fonts this command will use the `\accent` primitive to add an acute to the `e`; however if you are using T1 fonts, the existing  $\acute{e}$  will be accessed directly. Note however that the position of  $\acute{e}$  in the output encoding (T1) is typically *different* from the position of the character in the input encoding used.

`inputenc` Specifies that an 8-bit input encoding is being used. A package option should always be used which sets up the default encoding. The currently available options include `latin1`, `latin2`, `ansinew`, `cp437`, `cp437de`, `applemac`. (The two IBM codepage 437 variants differ in just one slot, the former uses  $\beta$ , the latter uses  $\beta$ .)

<sup>3</sup>One should be able to find details of such local variants in the famous ‘local guide’.

So typical usage (to specify ISO Latin-1 input conventions) would be:

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

`fontenc` Specifies the default output encoding for text fonts. Currently the available options are OT1 and T1. So to specify that fonts in the the T1 (Cork) encoding be used in the document one would declare:

```
\usepackage[T1]{fontenc}
```

## 7.2 Remaining Packages in the Base Distribution

`alltt` Defines the `alltt` environment, similar to `verbatim` except that `\`, `{` and `}` retain their usual  $\TeX$  meanings.

`doc` The package defining the commands used for documenting all the  $\LaTeX$  code in the distribution.

`shortverb` This package (really a small part of the `doc` package) defines the `\MakeShortVerb` command that allows shorthands like `|\foo|` instead of `\verb|\foo|`. This is very convenient if you are documenting  $\TeX$  or some other situation where you need to make a lot of use of short sections of verbatim text.

`exscale` For mainly historical reasons  $\LaTeX$  always uses the math extension font (used for brackets and sum and integral signs etc.) at the same size, whatever the current font size. This package modifies this behaviour so that magnified fonts are used at larger sizes. At the same time it makes the plain  $\TeX$  commands `\big`, `\bigg` etc., work as expected in conjunction with  $\LaTeX$  size commands.

`flafter`  $\LaTeX$  floats such as the `figure` and `table` environment can float *up* to the top of the current page. This means that it is possible that the figure appears before its first reference. Some publisher's styles do not allow this. `after` redefines the float placement algorithm so that a float never appears before its position in the source file, so by using this package, and placing the `figure` environment after the first reference to the figure, one can ensure that a figure will appear after the reference.

`graphpap` The `\graphpaper` command produces a grid for use in the `picture` environment.

`ifthen` Provides an 'if... then... else...' programming construct for use in  $\LaTeX$  packages. Many of the examples in *The  $\LaTeX$  Companion* assume this package has been loaded.

`makeindx` Implements support for generating an index.

`pict2e` This package produces an error message to say that it has not been written. The documentation in the `pict2e.dtx` source file suggests alternative packages for extending `picture` mode, `pspicture` (D. P. Carlisle), `pspic` (K. K. Thorup), and the very powerful `pstricks` (T. V. Zandt).

`showidx` This causes the argument of each `\index` command to be printed on the page where it occurs. See also `idx.tex` described below.

`syntonly` Used to process a document without typesetting it. On some systems this speeds things up considerably, and so may (possibly) be useful while debugging documents.

`tracefmt` This allows you to control how much information about  $\LaTeX$ 's font loading is displayed.

`latexsym` Loads the special  $\LaTeX$  symbol font and then defines commands such as `\Box` that use this font. These commands were defined by default in  $\LaTeX$  2.09.

`newlfont` Defines 'old' font commands to act in the 'new' way. For example it makes `\rm` essentially equivalent to `\rmfamily`. This package is not now recommended but is distributed so old documents written using the  $\LaTeX$  2.09 version of this package still work.

`oldlfont` A companion to `newlfont`. This package is only to be used for old documents that used the  $\LaTeX$  2.09 package of the same name.

## 8 Font Definition Files

Unpacking the  $\LaTeX$  distribution creates dozens of 'font definition files' with extension `.fd` from their documented sources (with extension `.fdd`). These map the internal  $\LaTeX$  model of fonts on to the external file names as used on your system. Normally you never need to load these explicitly into a  $\LaTeX$  document and they will not be considered in detail here except to say that if you obtain some new fonts from the  $\TeX$  archives, make sure to also get the related `fd` files, and install them where  $\LaTeX$  can find them.

## 9 Makeindex Styles

The distribution includes three styles (with extension `.ist`) for the `makeindex` index generator. They modify the `makeindex` defaults so as to work with the special requirements of the `doc` package.

`gind` Produces indices of command definition and use.

`gglo` Produces 'change log' entries (using the  $\LaTeX$  `\glossary` command rather than `\index`).

`source2e` This style is produced only if the  $\LaTeX$  document `source2e.tex` is processed. It is almost identical to `gind.ist` but defines 'I' to be in the series 'I-J-K' rather than 'I-II-III'. This is needed for the numbering conventions used in that document.

## 10 Miscellaneous Utilities and Files

`idx.tex` Print out index entries in your document.

`lablst.tex` Generate list of labels used in a document. You may prefer instead to have the labels show up in the margins of your drafts, in which case use the `showkeys` package from the 'tools' collection to be described later in this 'tour'.

`ltxcheck.tex` This 'document' should always be processed after  $\LaTeX$  has been installed. It pro-

duces no output but checks that various components of the system are configured correctly for your machine type.

`nfssfont.tex` Test file for testing a font. A more extensive font test is available if you use the `fontsmpl` package from the ‘tools’ collection.

`testpage.tex` Test file for checking the accuracy of a printer. This is particularly useful to see if you need to specify any offsets to your printer driver to ensure that the printed text is correctly positioned on the paper.

`Makefile.unx` A very simplistic template ‘Makefile’ for installing the L<sup>A</sup>T<sub>E</sub>X base distribution under UNIX. Many UNIX T<sub>E</sub>X distributions come with far more suitable installation procedures. For example the excellent ‘teT<sub>E</sub>X’ distribution allows you to install T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, metafont, dvips, xdvi, and a host of other utilities and fonts just by typing `sh install.sh`.

`latex209.def` This file is loaded whenever a document beginning with `\documentstyle` is seen. It forces L<sup>A</sup>T<sub>E</sub>X into ‘2.09 compatibility mode’ which is a slow, but a fairly accurate, emulation of the old version of L<sup>A</sup>T<sub>E</sub>X. This enables old documents to be processed under the current system.

## 11 Conclusion

The base distribution described here is only really a bare minimum L<sup>A</sup>T<sub>E</sub>X installation. Most sites would want to install all of the L<sup>A</sup>T<sub>E</sub>X files from the CTAN `latex/packages` directory, and some of the contributed packages as well. These other files will be covered in future installments of this ‘tour’. Part 2 will cover the `tools` and `graphics` distributions. Part 3 will cover `babel`, `mfnfss` and `psnfss`. Part 3 will cover the AMS L<sup>A</sup>T<sub>E</sub>X distributions `amsfonts` and `amslatex`. If I am still feeling keen, later parts may venture out to describe some of the more popular contributed packages.