Modifying LaTeX

12 December 1995

Contents

Abstract

This document was produced in response to suggestions that the modification and distribution conditions for the files constituting the New Standard LaTeX system should be similar to those implied by Version 2 of the GNU General Public Licence, as published by the Free Software Foundation.

Introduction

This article describes the principles underlying our policy on distribution and modification of the files comprising the LaTeX system. It has been produced as a result of detailed discussions of the issues involved in the support and maintenance of a widely distributed document processing system used by diverse people for many applications. These discussions have involved users, maintainers of installations that support LaTeX and various types of organisations that distribute it. The discussions are continuing and we hope that the ideas in this article will make a useful contribution to the debate.

Our aim is that LaTeX should be a system which can be trusted by users of all types to fulfill their needs. Such a system must be stable and well-maintained. This implies that it must be reasonably easy to maintain (otherwise it will simply not get maintained at all). So here is a summary of our basic philosophy:

> We believe that the freedom to rely on a widely-used standard for document interchange and formatting is as important as the freedom to experiment with the contents of files.
>
> We are therefore adopting a policy similar to that which Donald Knuth applies to modifications of the underlying TeX system: that certain files, together with their names, are part of the system and therefore the contents of these files should not be changed unless the following conditions are met:
>
> - they are clearly marked as being no longer part of the standard system;
> - the name of the file is changed.

The system

In developing this philosophy, and the consequent limitations on how modifications of the system should be carried out, we were heavily influenced by the following facts concerning the current widespread and wide-ranging uses of the LaTeX system.

1. LaTeX is not just a document processing system; it also defines a language for document exchange.
2. The standard document class files, and some other files, also define a particular formatting of a document.
3. The packages that we maintain define a particular document interface and, in some cases, particular formatting of parts of a document.
4. The interfaces between different parts of the LaTeX system are very complex and it is therefore very difficult to check that a change to one file does not affect the functionality of both that file and also other parts of the system not obviously connected to the file that has been changed.

This leads us to the general principle that:

> with certain special exceptions, if you change the contents of a file then the changed version should have a different file name.

We certainly do not wish to prevent people from experimenting with the code in different ways and adapting it to their purposes. However, we are concerned that any distribution of modifications to the code should be very clearly identified as not being a part of the standard distribution. The exact wording and form of the distribution conditions is thus something that is flexible, but only within the constraint of keeping LaTeX as a standardised, reliable product for the purposes described above: the exchange and formatting of documents.

Some examples

Here we elaborate the arguments that have led us to the above conclusion.

Separate development considered harmful!

In many fields, the use of LaTeX as a language for communication is just as important as its capacity for fine typesetting; this is a very important consideration for a large population of authors, journal editors, archivists, etc.

Related to this issue of portability is the fact that the file names are part of the end-user syntax.

As a real example, the LaTeX 'tools' collection contains the package 'array.sty'. A new user-level feature was added to this file at the end of 1994 and a document using this feature can contain the line:

```
\usepackage{array}[1994/10/16]
```

By supplying the optional argument, the document author is indicating that a version of the file `array.sty` dated no earlier than that date is required to run this document without error.

This feature would be totally worthless if we were to allow an alternative version of the array package to be distributed under the same name since it would mean that there would be in circulation files of a later date, but without the new feature. If the document were processed using this 'alternative array' then it would certainly produce 'undefined command' errors and would probably not be processable at all.

What's in a file-name?

In a pure markup language, such as SGML, it is reasonably clear that control over the final presentation lies with the receiver of a document and not with the author.

However, the way that LaTeX is often used in practice means that most people (at least when using the standard classes and packages) expect the formatting to be preserved when they send the document to another site.

For example, suppose, as is still the most common use of LaTeX in publishing, you produce a document for 'camera-ready-copy' using the class 'article' and that you carefully tune the formatting by, for example, adding some explicit line breaks etc, to ensure that it fits the 8 page limit set by the editor a journal or proceedings.

It then gets sent to the editor or a referee who, without anyone knowing, has a non-standard version of the class file 'article' and so it then runs to 9 pages. The consequence of this will, at the least, be a lot of wasted time whilst everyone involved works out what has gone wrong; it will probably also lead to everyone blaming each other for something which was in fact caused by a misguided distribution policy.

It should also be noted that, for most people, the version of the class file 'article' that gets used is decided by a site maintainer or the compilers of a CD-ROM distribution. To most users, the symbols `a r t i c l e` in:

```
\documentclass{article}
```

are just as much part of LaTeX's syntax as are the symbols `1 2 p t` in:

```
\hspace{12pt}
```

Thus they should both define a standard formatting rather than sometimes producing 1 more page or a 5pt larger space.

Users rely on the fact that the command (or menu item) 'LaTeX' produces a completely standard LaTeX, including the fact that 'article' is the 'standard article'. They would not be at all happy if the person who installed and maintains LaTeX for them were allowed to customise 'article' every second day so as (in her or his opinion) to improve the layout; or because another user wanted to write a document in a different language or typeset one with different fonts.

TeX itself

We have modelled our policies on those of the TeX system since this has for some time now been widely acknowledged as a very stable and high quality typesetting system.

The distribution policy set up by Donald Knuth for TeX has the following features:

- There is a clearly specified method for changing parts of the software by the use of 'change files'.

- Although arbitrary changes are allowed, the resulting program can be called TEX only if its functionality is precisely the same as that of TEX (i.e. neither less nor more) in all important areas.
- There are many files in the system that cannot be changed at all (without changing the name): examples are the file `plain.tex` and the files associated with fonts, including the Metafont source files.

Maintaining complexity

Our experience of maintaining LATEX has shown us just how complex are the interactions between different parts of the system.

We have therefore, with lots of help from the bug reports you send in, developed a large suite of test files which we run to check the effects of every change we make. A non-negligible percentage of these test runs give unexpected results and hence show up some unexpected dependency in the system.

Some assurances

We are certainly not attempting to stop people reformatting LATEX documents in any way they wish. There are many ways of customising incoming documents to your personal style that do not involve changing the contents of LATEX's standard files; indeed, this freedom is one of the system's many advantages. The simplest way to achieve this is to replace

`\documentclass{article}` by `\documentclass{myart}`

Nor do we wish to discourage the production of new packages improving on the functionality or implementation of those we distribute. All we ask is that, in the best interests of all LATEX users, you give your superbly improved class or package file some other name.

Configuration possibilities

The standard LATEX system format can be configured in several ways to suit the needs and resources of an installation. For example, the loading of fonts and font tables can be customised to match the font shapes, families and encodings normally used in text mode. Also, by producing the appropriate font definition files, the font tables themselves can be set up to take advantage of the available fonts and sizes. The loading of hyphenation patterns can be adjusted to cover the languages used; this has to be done as part of making the format since this is the only stage at which patterns can be loaded.

A complete list of these configuration possibilities can be found in the distributed guide *Configuration options for* LaTeX 2_ε (`cfgguide.tex`). However, as it says there, the number of configuration possibilities is strictly limited; we hope that having read this far you will appreciate the reasons for this decision. One consequence of this is that there is no provision for a general purpose configuration file, or for adding extra code just before the `\dump` of the format file.

This was a deliberate decision and we hope that everyone (yes, that includes you!) will support its intent. Otherwise there will be a rapid return to the very situation, of several incompatible versions of LaTeX 2.09, that originally prompted us to produce LaTeX 2_ε: the new, and *only*, 'Standard LaTeX'. This will make LaTeX unmaintainable and, hence, unmaintained (by us, at least).

> Therefore you should not misuse the configuration files or other parts of the distribution to produce non-standard versions of LaTeX.

Some of the allowed configurations can result in a system that can produce documents that are no longer 'formatting compatible'; for example, the use of different default fonts will most likely produce different line and page breaks. If you do produce a system that is configured in such a way that it is not 'formatting compatible' then you should consider carefully the needs of users who need to create portable documents. A good way to provide for their needs is to make available, in addition, a standard form of LaTeX without any 'formatting incompatible' customisations.

Modification conditions

It is possible that you need to produce a document processing system based on standard LaTeX but with functionality that cannot be implemented by using the approved configuration files and complying with the restriction on the code that is allowed in them. In other words, you may need a system which is sufficiently distinct from Standard LaTeX that it is not feasible to do this simply by using the configuration options we provide or by producing new classes and packages.

If you do produce such a system then, for the reasons described above, you should ensure that your system is clearly distinguished from Standard LaTeX in every possible way, including the following.

1. Give your system a distinguished name, such as NS-TeX, which clearly distinguishes it from LaTeX.
2. Ensure that it contains no file with a name the same as that of a file in the standard distribution but with different contents. (If this is not possible then you must:

- ensure that files from the non-LaTeX system cannot be accidentally accessed whilst using a standard LaTeX;
- ensure that each file from the non-LaTeX system clearly identifies itself as a non-LaTeX file on the terminal and in the log file.)

3. Ensure that the method used to run your system is clearly distinct from that used to run Standard LaTeX; e.g. by using a command name or menu entry that is clearly not `latex` (or `LaTeX` etc).

4. Ensure that, when a file is being processed by your system, the use of non-standard LaTeX is clearly proclaimed to the user by whatever means is appropriate.

5. Ensure that what is written at the beginning of the log file clearly shows that your system has been used, and that it is not Standard LaTeX. See the file `cfgguide.tex` for how to achieve this.

6. Clearly explain to users that bug reports concerning your system should not be sent to the maintainers of Standard LaTeX.

Note to system administrators

If you install a non-standard (modified) version of LaTeX on a multi-user site then please, in addition, install Standard LaTeX and observe the conditions enumerated above, particularly 3.

What do you think?

We are interested in your views on the issues raised in this document. The best way to let us know what you think, and to discuss your ideas with others, is to join the `LaTeX-L` mailing list and send your comments there. To subscribe to this list, mail to:

`listserv@urz.uni-heidelberg.de`

the following one line message:

`subscribe LATEX-L` ⟨*your-first-name*⟩ ⟨*your-second-name*⟩