1 December 1994

Contents

1 Introduction

This document is an introduction to writing classes and packages for LaTeX, with special attention given to upgrading existing LaTeX 2.09 packages to LaTeX2e. The latter subject will also be covered in an article by Johannes Braams to be published in TUGboat.

1.1 Writing classes and packages for LaTeX2e

LaTeX is a document preparation system that enables the document writer to concentrate on the contents of their text, without bothering too much about the formatting of it. For example, chapters are indicated by n

Chapter 1

<title>

nther than by selecting 18pt bold.

The file that contains the information about how to turn logical structure (like in

Chapter 2

n

j) into formatting (like i18pt bold ragged rightj) is a *document class*. In addition, some features (such as colour or included graphics) are independent of the document class and these are contained in *packages*.

One of the largest differences between LaTeX 2.09 and LaTeX2e is in the commands used to write packages and classes. In LaTeX 2.09, there was very little support for writing n.styn files, and so writers had to resort to using low-level commands.

LaTeX2e provides high-level commands for structuring packages. It is also much easier to build classes and packages on top of each other, for example writing a local technical report class based on narticlen.

0.2 Overview

This document contains an overview of how to write classes and packages for LaTeX. It does *not* introduce all of the commands necessary to write packages: these can be found in either or . But it does describe the new commands for structuring classes and packages.

Section ? contains some general advice about writing classes and packages. It describes the difference between classes and packages, the command naming conventions, the use of ndocn and ndocstripn, how TeXjs primitive file and box commands interact with LaTeX. It also contains some hints about general LaTeX style.

Section ? describes the structure of classes and packages. This includes building classes and packages on top of other classes and packages, declaring options and declaring commands. It also contains example classes.

Section ? lists the new class and package commands.

Section ? gives detailed advice on how to upgrade existing LaTeX 2.09 classes and packages to LaTeX2e.

0.3 Further information

For a general introduction to LaTeX, including the new features of LaTeX2e, you should read by Leslie Lamport [A\s\do5(L)La94].

A more detailed description of the new features of LaTeX, and an overview of more than 150 packages, is to be found in by Michel Goossens, Frank Mittelbach and Alexander Samarin [A\s\do5(G)MS94].

The LaTeX system is based on TeX, which is described in *The* TeX*book* by Donald E. Knuth [A\s\do5(D)EK91].

There are a number of documentation files which accompany every copy of LaTeX. A copy of LaTeX *News* will come out with each six-monthly release of LaTeX, and is found in the files nltnews*.texn. The authorjs guide describes the new LaTeX document features; it is in nusrguide.texn. The guide describes the LaTeX font selection scheme for class- and package-writers; it is in nfntguide.texn.

We are gradually turning the source code for LaTeX into a LaTeX document . This document includes an index of LaTeX commands and can be typeset from nsource2e.texn.

For more information about TeX and LaTeX, please contact your local TeX Users Group, or the international TeX Users Group, P. O. Box 869, Santa Barbara, CA 93102-0869, USA, Fax: +1 805 963 8358, E-mail: tug@tug.org.

1 Writing classes and packages

This section covers some general points concerned with writing LaTeX classes and packages.

1.1 Old versions

If you are upgrading an existing LaTeX 2.09 style file then we recommend freezing the 2.09 version and no longer maintaining it. Experience with the various dialects of LaTeX which existed in the early 1990js suggests that maintaining packages for different versions of LaTeX is almost impossible. It will, of course, be necessary for some organisations to maintain both versions in parallel for some time but this is not essential for those packages and classes supported by individuals: they should support only the new standard LaTeX2e, not obsolete versions of LaTeX.

1.2 Using docstrip and doc

If you are going to write a large class or package for LaTeX then you should consider using the ndocn software which comes with LaTeX. LaTeX classes and packages written using this can be processed in two ways: they can be run through LaTeX, to produce documentation; and they can be processed with ndocstripn, to produce the n.clsn or n.styn file.

The ndocn software can automatically generate indexes of definitions, indexes of command use, and change-log lists. It is very useful for maintaining and documenting large TeX sources.

The documented sources of the LaTeX kernel itself ,and of the standard classes, etc, are ndocn documents; they are in the n.dtxn files in the distribution. You can, in fact, typeset the source code of the kernel as one long document, complete with index, by running LaTeX on nsource2e.texn.

For more information on ndocn and ndocstripn, consult the files ndocstrip.dtxn, ndoc.dtxn, and . For examples of its use, look at the n.dtxn files.

1.3 Is it a class or a package?

The first thing to do when you want to put some new LaTeX commands in a file is to decide whether it should be a *document class* or a *package*. The rule of thumb is:

> If the commands could be used with any document class, then make them a package; and if not, then make them a class.

There are two major types of class: those like narticlen, nreportn or nlettern, which are free-standing; and those which are extensions or variations of other classesnfor example, the nprocn document class, which is built on the narticlen document class.

Thus, a company might have a local nownletn class for printing letters with their own headed notepaper. Such a class would build on top of the existing nlettern class but it cannot be used with any other document class, so we have nownlet.clsn rather than nownlet.styn.

The ngraphicsn package, in contrast, provides commands for including images

into a LaTeX document. Since these commands can be used with any document class, we have ngraphics.styn rather than ngraphics.clsn.

1.4 Command names

LaTeX has three types of command.

There are the author commands, such as n

2 n

,nnd n*n: most of these have short names, all in lower case.

There are the class and package writer commands: most of these have long mixed-case names such as the following.

```
\InputIfFileExists \RequirePackage \
PassOptionsToClass
```

Finally, there are the internal commands used in the LaTeX implementation, such as ntempcntan, nifnextcharn and nehan: most of these commands contain n@n in their name, which means they cannot be used in documents, only in class and package files.

Unfortunately, for historical reasons the distinction between these commands is often blurred. For example, nn is an internal command which should only be used in the LaTeX kernel, whereas n@nen is the constant ?1 and could have been nn.

However, this rule of thumb is still useful: if a command has n@n in its name then it is not part of the supported LaTeX languagenand its behaviour may change in future releases! If a command is mixed-case, or is described in , then you can rely on future releases of LaTeX2e supporting the command.

2.1 Box commands and colour

Even if you do not intend to use colour in your own documents, by taking note of the points in this section you can ensure that your class or package is compatible with the ncolorn package. This may benefit people using your class or package who have access to colour printers.

The simplest way to ensure icolour safetyj is to always use LaTeX box commands rather than TeX primitives, that is use nn rather than n, nn rather than nn and n or nn rather than nn. The LaTeX

box commands have new

options which mean that they are now as powerful as the TeX primitives.As an example of what can go wrong, consider that in n<text>n the font is restored just *before* the nn, whereas in the similar looking construction n <text>n the colour is restored just *after* the final n